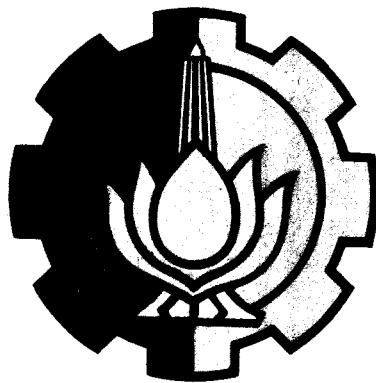
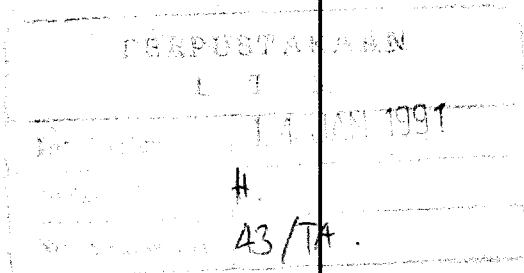


3886 / IIS / H / 91 V

# PERENCANAAN DAN PEMBUATAN STAVOLT ELEKTRONIK YANG TERPROGRAM DENGAN MIKROPROSESSOR Z-80



PSE  
621.391 6  
Sas  
P-1  
1990

Disusun oleh :

Cornelius Damar Sasongko

285 220 0363

BIDANG STUDI KOMPUTER  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
S U R A B A Y A  
**1990**

# **PERENCANAAN DAN PEMBUATAN STAVOLT ELEKTRONIK YANG TERPROGRAM DENGAN MIKROPROSESSOR Z-80**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Elektro**

**P a d a**

**Bidang Studi Komputer**

**Jurusan Teknik Elektro**

**Fakultas Teknologi Industri**

**Institut Teknologi Sepuluh Nopember**

**S u r a b a y a**

**Mengetahui / Menyetujui**

**Dosen Pembimbing**



**DR. Ir. SOEPENO DJANALI**

**S U R A B A Y A**

**NOPEMBER 1990**

# ABSTRAK

---

Sistem pengontrolan secara otomatis kestabilan tegangan listrik yang ada di dalam lingkungan perkantoran dan perumahan terasa perlu untuk diterapkan pada era komputer saat ini.

Pokok pengontrolan dalam pembahasan tugas akhir ini adalah waktu untuk mensuplay tegangan yang dibutuhkan oleh komputer seestabil mungkin . Peralatan yang dikontrol yang paling utama adalah komputer , disamping peralatan yang lain seperti misalnya televisi dan lampu lampu . Display led akan ditampilkan selama mikroprosesor melakukan proses pengontrolan setiap peralatan di atas.

Komponen utama yang digunakan adalah mikroprosesor Z-80 beserta komponen - komponen penunjang lainnya yang membentuk suatu minimum sistem. Komponen ADC 0808, triac . Perangkat lunak yang digunakan adalah bahasa Assembly Z-80 yang berfungsi untuk mengatur proses kerja sistem secara keseluruhan.

# KATA PENGANTAR

---

Dengan mengucapkan syukur kehadiran ALLAH yang telah melimpahkan rahmat dan karunianya sehingga penulis dapat menyelesaikan tugas akhir dengan judul :

" Perencanaan dan Pembuatan Stavolt Elektronik yang terprogram dengan mikroprosesor Z-80 "

Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana teknik Elektro bidang studi komputer pada Fakultas teknologi Industri Institut Teknologi Sepuluh November Surabaya.

Tak Pelak lagi kata pengantar ini kemudian menjadi semacam daftar panjang ucapan terima kasih. Sebab untuk menjadikan masalah tadi sebagai topik dan menuliskannya dalam bentuk tugas akhir , bantuan dari banyak orang sangatlah mutlak harus ada.

Yang pertama ucapan terima kasih tentu ditujukan oleh Bapak DR. Ir. Soepeno Djanali, dosen pembimbing penulisan yang sangat membantu terselesaikannya tugas akhir ini. Kemudian tak kalah besarnya ucapan terima kasih kepada Ir. Surya Perdana, yang telaten melayani diskusi yang mulanya tak berujung pangkal karena keterbatasan pengetahuan penulis.

Tugas akhir ini tak akan terwujud tanpa dukungan moral Mama, Bude Isma'oan dan Bude Sastro adik, R Danisworo , dan GM Saksono yang telah membantu dengan sepenuh hati dan juga teman teman di Lab B-201 ,Linggo ,kartolo ,Raden ,Peltu ,Tikno, Paidi ,Ableh.

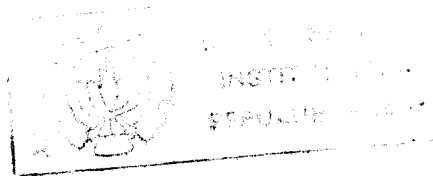
Yang terakhir dan mungkin paling penting adalah dukungan semangat tak terhingga dari Theresia Niken Woelandari yang setia mengikuti kebingungan pemikiran penulis sejak awal sampai akhir tugas akhir ini .

Surabaya , 7 Oktober 1990

CD SASONGKO

# D A F T A R    I S I

BAB	HALAMAN
JUDUL .....	1
PENGESAHAN .....	11
ABSTRAK .....	111
KATA PENGANTAR .....	1v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	viii
DAFTAR TABEL .....	x
I. PENDAHULUAN .....	1
I. 1. LATAR BELAKANG .....	1
I. 2. PERMASALAHAN .....	2
I. 3. METODOLOGI .....	2
I. 4. LANGKAH - LANGKAH PEMBAHASAN .....	3
II. TEORI PENUNJANG .....	4
II. 1. UNIT FUNGSIONAL SISTEM MIKROPROSESOR .....	4
II. 1. 1. ARSITEKTUR SISTEM 3 BUS .....	5
II. 1. 1. 1. SISTEM ADDRESS BUS DAN FUNGSIONYA .....	6
II. 1. 1. 2. SISTEM DATA BUS DAN FUNGSIONYA .....	6
II. 1. 1. 3. SISTEM KONTROL BUS DAN FUNGSIONYA .....	7
II. 1. 2. ADDRESS DECODING DAN BUFFERING .....	7
II. 1. 2. 1. CARA - CARA DECODING .....	8



## BAB

## HALAMAN

II. 1. 2. 2. IC DECODER SN74LS138 .....	9
II. 1. 3. MEMORY .....	10
II. 1. 3. 1. READ ONLY MEMORY .....	12
II. 1. 3. 1. 1. KARAKTERISTIK EPROM .....	14
II. 1. 3. 1. 2. MEMBACA DATA DARI ROM .....	15
II. 1. 3. 1. 3. INTERKONEKSI ANTARA Z-80 DENGAN EPROM .....	15
II. 1. 3. 2. RANDOM ACCESS MEMORY .....	16
II. 1. 3. 2. 1. STATIC RAM .....	17
II. 1. 3. 2. 2. DYNAMIC RAM .....	19
II. 1. 3. 2. 3. MEMBACA DATA DARI RAM .....	19
II. 1. 3. 2. 4. MENULIS DATA KE RAM .....	20
II. 2. MIKROPROSESOR Z-80 .....	21
II. 2. 1. ARSITEKTUR MIKROPROSESOR Z-80 .....	22
II. 2. 1. 1. AKUMULATOR (A) DAN FLAG (F) ...	23
II. 2. 1. 2. SPECIAL PURPOSE REGISTER .....	25
II. 2. 1. 3. GENERAL PURPOSE REGISTER .....	27
II. 2. 2. ARITHMATIC LOGIC UNIT .....	27
II. 2. 3. PIN - PIN PADA MIKROPROSESOR Z-80 .....	28
II. 3. UNIT INPUT - OUTPUT .....	32
II. 3. 1. PROGRAMABLE PERIPHERAL INTERFACE 8255 .....	33
II. 3. 1. 1. KETERANGAN PIN PADA 8255 .....	35
II. 3. 1. 2. CARA MENGHUBUNGKAN Z-80 DENGAN PPI 8255 .....	36
II. 3. 1. 3. REGISTER PADA 8255 .....	38

BAB	HALAMAN
II. 3. 1. 4. PPI 8255 PADA MODE 0 .....	40
II. 3. 2. ANALOG TO DIGITAL CONVERTER .....	41
III. PERENCANAAN .....	45
III. 1. BLOK DIAGRAM ALAT YANG DIRENCANAKAN .....	45
III. 2. CARA KERJA ALAT .....	45
III. 3. SISTEM MIKROPROSESOR .....	46
III. 3. 1. RANGKAIAN OSILATOR .....	46
III. 3. 2. RANGKAIAN RESET .....	47
III. 3. 3. RANGKAIAN DEKODER UNTUK MEMORY DAN I/O .....	48
III. 3. 3. 1. RANGKAIAN DEKODER UNTUK MEMORY .....	48
III. 3. 3. 2. RANGKAIAN DEKODER UNTUK I/O .....	48
III. 4 RANGKAIAN ADC .....	52
III. 5. RANGKAIAN I/O .....	52
III. 6. RANGKAIAN DRIVER .....	60
IV. PEMBUATAN DAN PENGUKURAN ALAT .....	59
IV. 1. PEMBUATAN ALAT .....	59
IV. 2. PENGUKURAN ALAT .....	60
IV. 2. 1. PENGUKURAN UNTUK FREKUENSI CLOCK Z-80 .....	60
V. KESIMPULAN .....	61
DAFTAR PUSTAKA .....	62
LAMPIRAN A : SOFTWARE LISTING .....	63
LAMPIRAN B : DATA SHEET .....	66
USULAN TUGAS AKHIR .....	75



# I D A I F T A R      G A M B A R

GAMBAR	HALAMAN
2-1      SISTEM MIKROPROSESOR STANDARD .....	04
2-2      3 BUS SISTEM ARSITEKTUR .....	05
2-3      IC DEKODER 74LS138 .....	10
2-4      BLOK DIAGRAM EPROM .....	13
2-5      DIAGRAM WAKTU PENGAMBILAN DATA DARI EPROM .....	15
2-6      HUBUNGAN ANTARA Z-80 DENGAN EPROM 2716 .....	17
2-7      READ ACCESS TIME .....	18
2-8      WRITE ACCESS TIME .....	18
2-9      DIAGRAM WAKTU PEMBACAAN DATA DARI RAM .....	19
2-10     DIAGRAM WAKTU PENULISAN DATA KE RAM .....	21
2-11     BLOK DIAGRAM ARSITEKTUR INTERNAL Z-80 .....	22
2-12     REGISTER - REGISTER PADA Z-80 .....	25
2-13     KONFIGURASI PIN Z-80 .....	29
2-14     8255 BLOK DIAGRAM .....	34
2-15     KONFIGURASI PIN - PIN 8255 .....	37
2-16     RANGKAIAN OR IC 74LS32 .....	38
2-17     HUBUNGAN ANTARA Z-80 DENGAN PPI 8255 .....	39
2-18     DEFINITION FORMAT 8255 .....	43
2-19     PENGUBAH ADC YANG MENGGUNAKAN COUPLER .....	44
3-1      BLOK DIAGRAM DARI SISTEM YANG DIRENCANAKAN .....	45
3-2      RANGKAIAN OSILATOR DAN RESET .....	48
3-3      PETA MEMORI .....	50

## GAMBAR

## HALAMAN

3-4	RANGKAIAN DEKODER UNTUK MEMORI .....	51
3-5	PETA PERALATAN I/O .....	51
3-6	RANGKAIAN DEKODER UNTUK PERALATAN I/O .....	54
3-7	RANGKAIAN I/O YANG MENGGUNAKAN PPI 8255 .....	56
3-8	RANGKAIAN ADC 0808 .....	58
3-9	RANGKAIAN DRIVER .....	61
4-1	FOTO ALAT YANG DIBUAT .....	67
4-2	BENTUK PULSA CLOCK DARI Z-80 CPU .....	68
4-4	BENTUK PULSA CLOCK DARI ADC 0808 .....	68

# DAFTAR TABEL

TABEL	HALAMAN
2-1 TABEL KEBENARAN IC DEKODER 74LS138 .....	11
2-3 LOKASI DAN FUNGSI REGISTER PADA 8255 .....	22
3-1 TABEL KEBENARAN DARI ALAMAT MEMORI .....	50
3-2 TABEL KEBENARAN DARI ALAMAT I/O .....	54

# IB A IB II

## IP IE N ID A IH U IL U A N

---

### I. 1. LATAR BELAKANG

Perkembangan teknologi dewasa ini demikian pesatnya, di mana - mana ditemui produk - produk teknologi yang modern. Kemajuan teknologi tersebut akan selalu mengarah ke unit yang lebih cepat, mutu yang lebih baik, kapasitas yang lebih besar dan volume yang sekecil mungkin, serta memanfaatkan energi yang serendah mungkin.

Sistem komputerisasi yang kini populer menyebabkan adanya perubahan - perubahan yang mendasar pada bidang - bidang perindustrian, perkantoran, pendidikan, kedokteran, rumah tangga dan lain sebagainya. Pengontrolan secara otomatis dengan menggunakan komputer telah menjadi bagian yang penting dan terpadu dari proses - proses dalam menangani pekerjaan baik di kantor, di pabrik - pabrik dan bahkan sampai ke rumah - rumah tangga dengan maksud untuk meningkatkan produktivitas, mempertinggi kualitas, menurunkan biaya produksi dan meniadakan pekerjaan - pekerjaan rutin yang membosankan yang harus dilakukan oleh manusia.

Pengaturan secara otomatis tegangan listrik yang ada di dalam suatu perkantoran dan perumahan untuk kepentingan supply tegangan dari beberapa peralatan ( Komputer , AC , lemari es, lampu , heater, oven, televisi dan lain - lain ) dengan

menggunakan mikroprosesor dirasakan sangat perlu untuk mengatasi masalah - masalah yang dihadapi apabila terjadi perubahan tegangan yang sangat tinggi .

## I. 2. PERMASALAHAN

Dalam tugas akhir ini akan dibuat suatu sistem yang dapat mengatur kestabilan tegangan listrik di rumah dan di kantor secara otomatis dengan menggunakan mikroprosesor Z-80 untuk pengontrolan sistemnya. Di mana yang menjadi pokok pengaturan adalah kestabilan tegangan sehingga peralatan yang digunakan dapat berfungsi dengan baik dan tidak cepat terjadi kerusakan.

## I. 3. METODOLOGI

Untuk merealisasikan apa yang sudah direncanakan, maka dibuat sistem mikroprosesor minimum Z-80 serta pembuatan interfacenya.

Perubahan tegangan yang bervariasi yang terdapat pada jaringan jala-jala PLN , dapat menyebabkan peralatan listrik yang mempunyai ketelitian yang tinggi akan menjadi cepat rusak.

ADC akan menerima input , dari tegangan jala-jala yang telah diturunkan sampai dibawah 5 volt. Data diambil untuk kemudian diolah sistem mikroprosesor , yang nantinya dibandingkan melalui software yang telah dibuat untuk , menyesuaikan dengan tegangan yang mendekati dengan tegangan inputnya.

Dengan mempelajari operasi sistem secara keseluruhan dari literatur - literatur maupun data book maka dibuatlah tugas akhir

ini berdasarkan teori - teori tersebut. Akhirnya berdasarkan petunjuk itu maka untuk menguji kebenarannya dibuatlah rangkaian-nya beserta dengan program yang diperlukan.

#### 1. 4. LANGKAH - LANGKAH PEMBAHASAN

Pembahasan tugas akhir ini dilakukan sebagai berikut :

Pada bab II akan dibahas teori penunjang dari komponen - komponen yang berhubungan dengan sistem mikroprosesor minimum seperti : mikroprosesor Z-80, memory, buffer, dekoder dan PPI 8255. Dibahas pula komponen - komponen yang berhubungan dengan I/O seperti ADC serta komponen - komponen yang menunjang rangkaian driver . Pada bab III dibahas mengenai perencanaan rangkaian serta pembuatan software sistem untuk mengontrol keseluruhan sistem sedangkan pada bab IV diberikan mengenai pengukuran dan pengujian dari alat yang dibuat. Dan akhirnya kesimpulan dari tugas akhir ini diberikan pada bab V.



# BAB III

## TEORI PENUNJANG

---

### II.1. UNIT FUNGSIONAL SISTEM MIKROPROSESOR

Suatu system mikroprosesor merupakan kesatuan dari beberapa unit fungsional , yaitu :

- Unit Pengolahan Pusat ( CPU , Central Prossesing Unit ) dalam sebuah chip tunggal yang di dalamnya terdapat ALU (Arithmethic Logic Unit) , register register internal dan unit kontrol (Control Unit ).
- Unit Memory
- Unit Input-output

Dalam sebuah mikroprosesor yang standard, hubungan antara unit-unit tersebut direalisasikan dalam arsitektur sistem 3 bus,

yaitu :

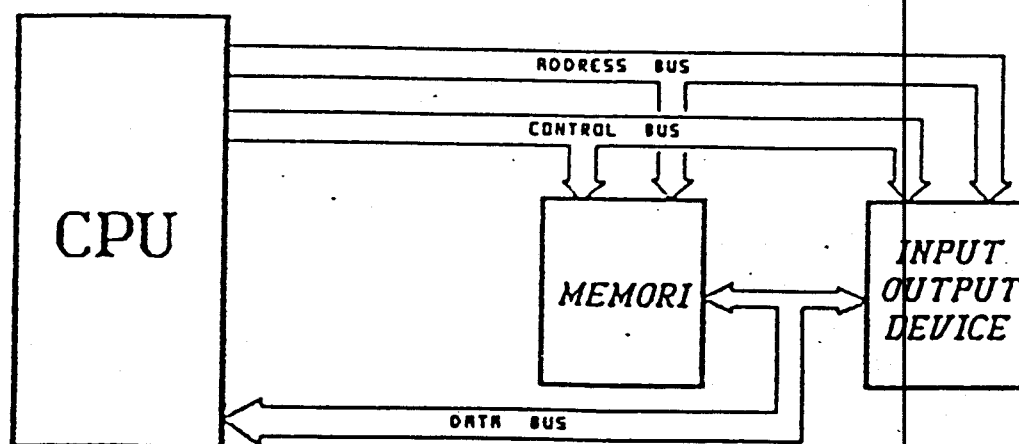
- Address bus
- Data bus
- Kontrol bus

#### II.1.1. ARSITEKTUR SISTEM 3 BUS

Yang tersebut dengan bus dalam mikroprosesor adalah kumpulan dari sinyal - sinyal komunikasi yang dikelompokkan berdasar - kan fungsinya. Pada mikroprosesor biasanya dibedakan dua macam bus, yaitu internal bus dan eksternal bus. Sinyal - sinyal yang terdapat pada mikroprosesor 8 bit dapat digolongkan menjadi 3 buah sistem bus, yakni :

1. Sistem Address Bus
2. Sistem Data Bus
3. Sistem Kontrol Bus

Gambar 2-1 menunjukkan gambar sistem mikroprosesor standard dengan arsitektur sistem 3 bus.



GAMBAR 2-1

#### SISTEM MIKROPROSESOR STANDARD

Oleh karena terdiri dari 3 buah sistem bus maka disebut 3 bus arsitektur.

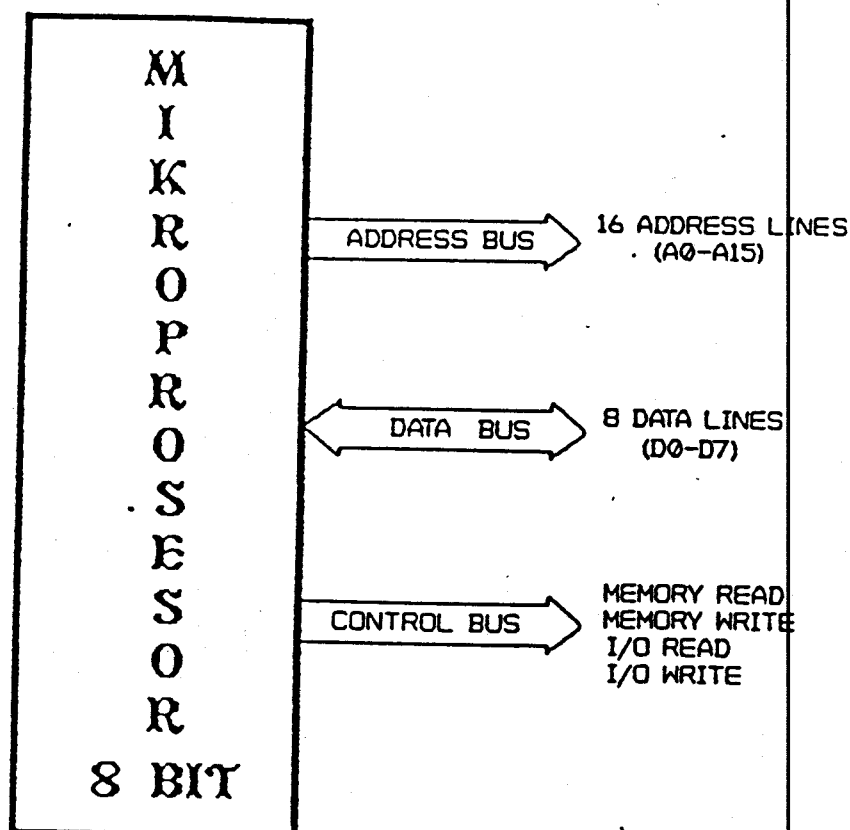
##### II. 1. 1. 1. SISTEM ADDRESS BUS DAN FUNGSINYA

Pada mikroprosesor 8 bit, sistem address bus merupakan pin output yang dapat berhubungan dengan komponen lain. Karena address bus merupakan pin output maka hanya mempunyai satu arah saja (unidirectional). Jumlah address bus menentukan besar kecilnya kemampuan mikroprosesor untuk berhubungan dengan lokasi memory. Misalnya mikroprosesor Z80 yang mempunyai 16 address bus (A0-A15), maka mikroprosesor tersebut mampu memberi alamat pada

memory sebesar  $2^{16} = 65.536$  lokasi.



Adapun blok diagram sistem mikroprosesor dengan 3 bus arsitektur dapat dilihat pada gambar 2-2.



GAMBAR 2-2

### 3 BUS SISTEM ARSITEKTUR

Setengah dari jumlah bit address ini yaitu delapan bit address yang mempunyai orde rendah digunakan untuk menentukan lokasi I/O. Jadi I/O hanya dapat mempunyai lokasi sebanyak  $2^8 = 256$  lokasi (dari 00h sampai FFh).

Sedangkan fungsi dari address bus adalah untuk menentukan alamat atau memilih suatu lokasi memory atau peripheral yang akan diajak berkomunikasi dengan mikroprosesor.

#### II. 1. 1. 2. SISTEM DATA BUS DAN FUNGSINYA

Sistem data bus merupakan in-out data dari suatu mikroprosesor. Jadi sifat dari data adalah dua arah (bidirectional), artinya data dapat dikirim dari memory ke sistem mikroprosesor ke sistem memory dan ke system I/O , dan sebaliknya dari sistem memory dan I/O ke mikroprosesor.

Pada waktu Write, data ditransfer dari mikroprosesor ke memory atau peripheral, sedang pada waktu Read, data ditransfer dari memory atau peripheral ke mikroprosesor. Meskipun data bus tidak dapat digunakan untuk mengirim dan menerima data . Jadi sinyal hanya dapat dikirimkan atau diterima pada satu arah saja pada suatu saat.

Fungsi dari data bus yaitu menyediakan jalur untuk pertukaran informasi data antara mikroprosesor dengan komponen luar dari mikroprosesor yang telah dipilih oleh address bus.

#### II. 1. 1. 3. SISTEM KONTROL BUS DAN FUNGSINYA

Sistem kontrol bus pada mikroprosesor mempunyai 4 (empat) buah sinyal yang tergabung dalam kontrol bus yang mempunyai fungsi sama yaitu sebagai kontrol pada mikroprosesor. Ke empat sinyal tersebut yaitu :

1. Memory Read (MEMR)
2. Memory Write (MEMW)
3. I/O Read (IOR)
4. I/O write (IOW)

Seperti halnya address bus , maka kontrol bus ini merupakan sinyal yang bersifat satu arah (unidirectional).

Bila mikroprosesor sedang membaca data dari peralatan I/O maka sinyal MEMR aktif, bila mikroprosesor menulis data ke memory maka sinyal MEMW aktif dan bila mikroprosesor membaca dari peralatan I/O maka sinyal IOR aktif serta bila mikroprosesor menulis data ke peralatan I/O maka sinyal IOW yang aktif.

Adapun fungsi dari sistem kontrol bus adalah menentukan hubungan kerja antara mikroprosesor dengan memory dan I/O, baik untuk kondisi Read maupun Write.

### II. 1. 2. ADDRESS DECODING

Address decoding diperlukan dalam merangkai suatu sistem mikroprosesor karena ada beberapa peralatan I/O yang dihubungkan ke mikroprosesor.

Untuk berkomunikasi dengan peralatan I/O maka diperlukan suatu cara agar mikroprosesor dapat memilih atau menentukan salah satu IC atau peralatan tertentu saja yang dapat berkomunikasi dengan mengaktifkan CS (Chip Select) nya, agar supaya tidak terjadi konflik akibat dari beberapa IC atau peralatan I/O yang aktif berkomunikasi dengan mikroprosesor pada saat yang bersamaan. Hal ini dapat dimungkinkan dengan suatu cara yang disebut dengan Address Decoding, dimana sinyal-sinyal pada address bus di-decode sedemikian rupa sehingga setiap kombinasi pada address bus akan menghasilkan satu sinyal pilih yang akan mengaktifkan salah satu IC atau peralatan yang sedang dipilih oleh mikroprosesor.

Output dari mikroprosesor terbatas. Misalnya pada mikroprosesor Z-80 maksimum beban yang diijinkan saat logika

"0" ( $I_{OL}$ ) adalah 1,8 mA.

#### II. 1. 2. 1. CARA-CARA DECODING

Ada beberapa cara yang dapat digunakan untuk men-decode suatu address sehingga dihasilkan sinyal pilih (chip select sinyal ), yaitu gating dan decoding.

##### - GATING

Gating adalah cara yang paling sederhana untuk men-decode suatu kombinasi address dengan menggunakan gate-gate logika seperti AND, OR dan NOT. Dengan metode gating ini setiap kombinasi address memerlukan satu rangkaian logika untuk mendapatkan satu sinyal pilih. Cara ini tidak efisien untuk sinyal pilih yang banyak.

##### - DECODING

Cara yang paling mudah dan efisien untuk decoding adalah dengan menggunakan decoder . Dengan penggunaan decoder , maka setiap  $n$  kombinasi address dapat diperoleh sebanyak  $2^n$  sinyal pilih , sehingga untuk sistem yang memerlukan banyak sinyal pilih biasanya dipakai cara decoding . Jika sinyal pilih yang merupakan output dari decoder langsung dihubungkan ke chip select suatu IC, maka disebut dengan partial decoding, sedangkan jika output tadi dipakai lagi sebagai chip select decoder berikutnya sehingga diperoleh suatu susunan decoder yang bertingkat maka disebut dengan fully decoding .

Ada beberapa macam decoder yang dapat digunakan dalam suatu sistem mikroprosesor untuk menentukan lokasi address memory dan I/O , antara lain :

1. 74LS138 (3-bit binary decoder)
2. 74LS139 (2 x 2-bit binary decoder )
3. 74LS242 (BCD to decimal decoder )

Dalam perencanaan tugas akhir ini menggunakan decoder 74ls138 untuk menentukan lokasi address EPROM, RAM, 6255 dan ADC 0808.

#### II. 1. 2. 2. IC DECODER SN74LS138

IC decoder SN74LS138 mempunyai 3 buah input untuk kombinasi address-nya dan 8 buah output untuk menghasilkan sinyal pilih yang sesuai . IC ini disebut 3 to 8 line decoder. Pada data sheet terlampir , 3 input tersebut adalah pin A , pin B dan pin C dimana pin C adalah Most Significant Bit (MSB) dan pin A adalah LSB (Least Significatnt bit ), sedangkan ke 8 outputnya adalah Y0 sampai dengan Y7. Jika pin pin enable diberi sinyal yang sesuai dengan tabelnya (G1 high , G2A dan G2B , kedua - duanya low), maka akan diperoleh kombinasi input dari bit-bit address akan menghasilkan salah satu output yang sesuai low.

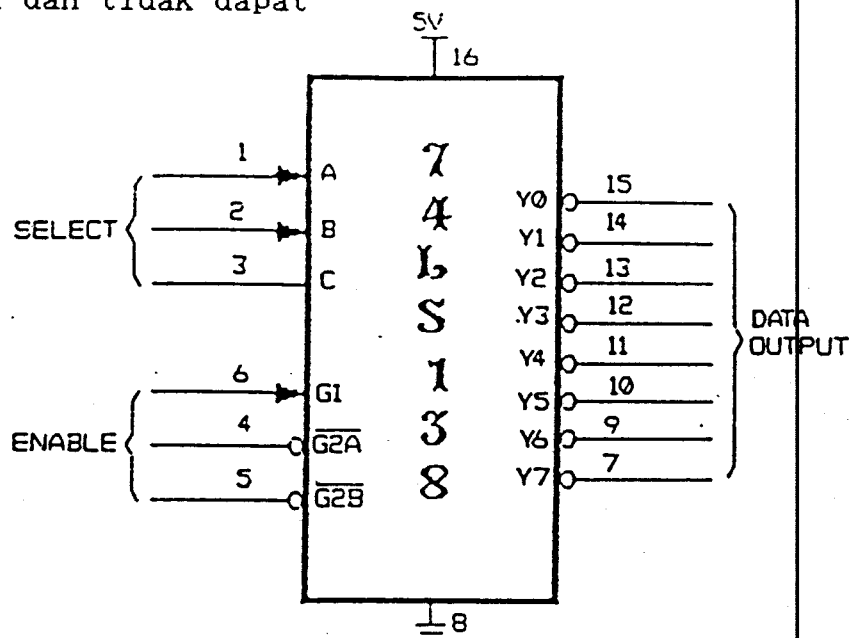
#### II. 1. 3. MEMORI

Di dalam suatu sistem mikroprosesor terdapat bagian yang sangat penting yang dinamakan memory. Memory merupakan tempat penyimpanan informasi yang dapat berupa data atau instruksi bagi suatu sistem mikroprosesor . Informasi dalam memory ada yang bersifat sementara (volatile) dan bersifat permanen(non volatile).

Memory sendiri terdiri dari beberapa jenis yaitu ROM(Read Only Memory ) dan RAM (Random Access Memory ).

### II. 1. 3. 1. READ ONLY MEMORY

Pada suatu sistem mikroprosesor diperlukan suatu memory yang menyimpan data untuk selamanya ( non volatile ). Dikatakan ROM karena informasi yang ada didalamnya hanya dapat dibaca saja dan tidak dapat



IC DEKODER 74LS138

TABEL 2-1

TABEL KEBENARAN IC DEKODER 74LS138

INPUTS		OUTPUTS										
ENABLE		SELECT										
G1	G2X	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	L	H	H	H	H	H
H	L	L	L	L	H	H	H	L	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H
H	L	L	L	L	H	H	H	H	L	H	H	H
H	L	L	L	H	H	H	H	H	L	H	H	H
H	L	L	L	L	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	L	H
H	L	L	L	L	H	H	H	H	H	H	H	L

diubah, sehingga ROM ini sangat berguna dalam suatu sistem karena dapat menginisialisasi semua peripheral hardware pada keadaan logika yang sesuai pada saat pertama kali sistem dinyalakan.

Ada beberapa jenis dari non volatile memory yang dapat digunakan dalam suatu sistem mikroprosesor yaitu :

1. Read Only Memory (ROM)
2. Programmable Read Only memory (PROM)
3. Erasable Programmable Read Only Memory (EPROM)
4. Electrically Alterable Read Only Memory (EAROM)

Pada ROM, informasi dimasukkan dalam memory chip oleh pabrik pembuatnya dan proses pemrograman oleh pabrik ini membutuhkan biaya yang besar , oleh karena itu ROM hanya dibuat untuk diisi dengan data tertentu yang sudah standard yang tidak berubah seperti character generator dengan standard ASCII ,dan lain sebagainya.

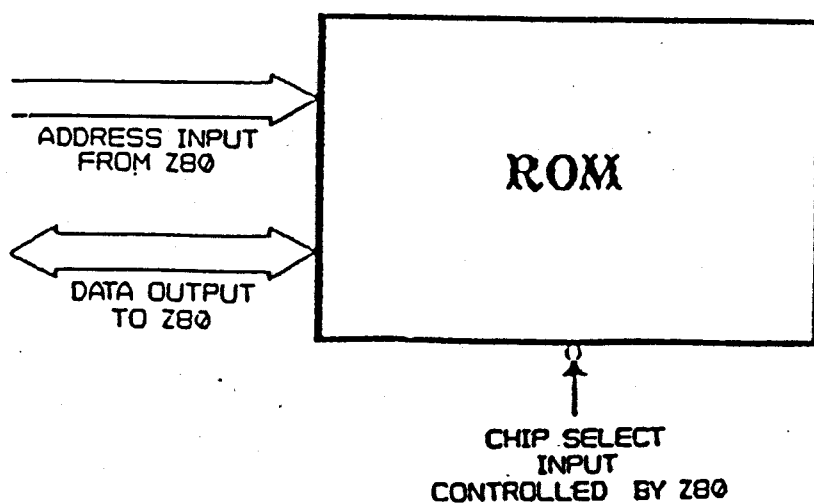
Pada PREOM, EPROM dan EAROM , data diprogramkan ke dalam memory chip oleh user dengan suatu alat yang dapat memberikan pulsa tegangan yang cukup tinggi ke dalam sel-sel memory chip. Pada ROM dan PROM informasi yang telah disimpan tidak dapat dihapus lagi ,sedangkan EPROM dan EAROM informasi yang telah disimpan masih dapat dihapus dengan memberikan sinar Ultra Violet dengan intensitas tertentu pada window dari IC pada EPROM atau dengan memberikan pulsa dengan tegangan yang cukup tinggi pada EAROM. Jenis ROM yang umum dipakai adalah EPROM. Jenis ROM yang umum dipakai adalah EPROM , dengan kapasitas penyimpanannya ada yang 1 KB (2708), 2 KB (2716), 4 KB (2732) , 8KB (2764) , 16 KB (271280) , dan lain-lain. Untuk EPROM yang mempunyai kapasitas

sebesar 2 Kb dan 4 Kb banyak sekali dipakai dalam rangkaian elektronik di pasaran.

#### II. 1. 3. 1. 1. KARAKTERISTIK EPROM

EPROM hanya dapat memberikan data pada mikroprosesor saja ( Read Only ), dengan memberikan input alamat dari EPROM tersebut . Blok diagram dari EPROM secara garis besar ditunjukkan pada gambar 2-6.

Banyaknya memory data yang dapat disimpan dalam suatu EPROM biasanya ditulis 2048 x 8 untuk 2 kilo-byte, atau 4096 x 8 untuk 4 kilo-byte. Deretan angka yang pertama, misalnya 2048, menunjukkan jumlah lokasi address yang dapat disimpan dalam EPROM sedangkan deretan yang kedua, misalnya 8, menunjukkan jumlah bit data paralel yang dapat diterima dari EPROM untuk setiap lokasi address . Untuk lokasi address sejumlah 2048 lokasi dibutuhkan 11 jalur address, karena kombinasi biner sejumlah 2048 atau dua pangkat sebelas.



GAMBAR 2-4 (1)

#### BLOK DIAGRAM EPROM

1) Coffron, James V., Practical Hardware Details for 8080, 8085, 286 and 386 Microprocessors system. New Jersey : Prentice-Hall Inc, 1981.



Dalam suatu EPROM terdapat suatu kaki yang disebut Chip Select (CS). Fungsi dari Chip Select ini adalah untuk mengaktifkan dan menonaktifkan data output dari EPROM. Dengan membuat Chip Select ini aktif, data memory akan dikeluarkan pada output data line sesuai dengan lokasi address yang diinputkan pada EPROM. Jika chip select ini tidak aktif, semua data output dalam keadaan tri state (High Impedance state).

#### II. 1. 3. 1. 2. MEMBACA DATA DARI ROM

Urut - urutan sinyal yang diperlukan untuk membaca data dari ROM adalah sebagai berikut:

1. CPU memberikan input address kepada pin pin ROM sesuai dengan lokasi yang akan dibaca.
2. Kemudian CPU menunggu untuk selang waktu tertentu (Read Access Time) berkisar antara 100 - 300 nano detik. Selang waktu ini diperlukan oleh rangkaian di dalam ROM untuk men-decode address yang diterima dan untuk menunggu data dari lokasi memory yang dituju mencapai data output line dari ROM.
3. Sinyal Chip Select diaktifkan, maka data akan dikeluarkan oleh EPROM pada data bus, kemudian CPU men-strobe data tersebut masuk kedalam internal register nya.
4. Kemudian sinyal chip select tidak diaktifkan untuk meniadakan data dari EPROM pada sistem data bus, sehingga alamat dan jalur data pada EPROM pada kea-

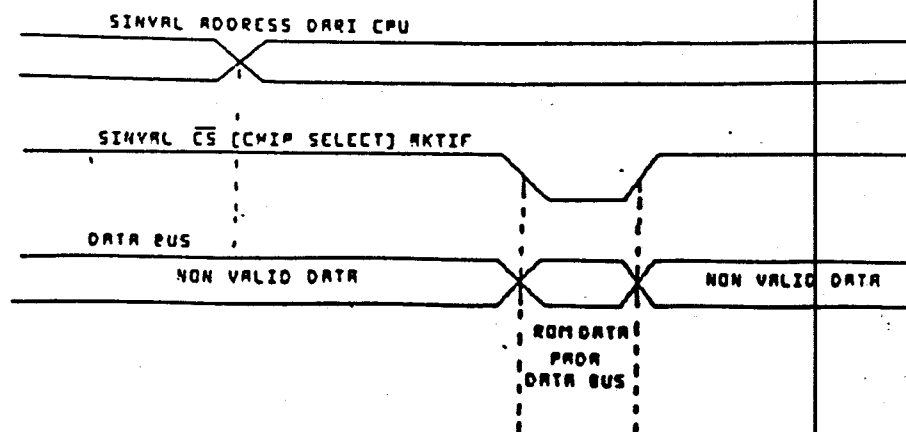
daan tri-state (impedansi tinggi).

Urutan sinyal - sinyal tersebut di atas selalu terjadi setiap saat CPU membaca ROM. Diagram waktu untuk urutan sinyal ditunjukkan dalam gambar 2-7.

#### II. 1. 3. 1. 3. INTERKONEKSI ANTARA Z-80 DENGAN EPROM

Hubungan antara Z-80 dengan EPROM 2716 ditunjukkan dalam gambar 2-6. Pada gambar 2-6 terlihat semua jalur alamat dan jalur data dari Z-80 masing - masing paralel dengan jalur data dan jalur alamat dari EPROM.

Chip select pada kaki 18 EPROM masih dalam keadaan terbuka. Kaki 20 EPROM disebut Output Enable. Dengan mengaktifkan kaki ini, data pada EPROM dapat dioutputkan pada bus data. Cara mengaktifkan output ini adalah dengan menghubungkan pada logika nol atau ke ground. Jika kaki ini tidak aktif, data pada memori tidak dapat keluar ke data bus walaupun chip selectnya diaktifkan.



GAMBAR 2-5(2)

DIAGRAM WAKTU PENGAMBILAN DATA DARI EPROM

### II. 1. 3. 2. RANDOM ACCESS MEMORY

Bila dibandingkan dengan ROM, maka RAM merupakan sistem memori yang menyimpan data hanya untuk sementara, karena bila power supply dimatikan maka data yang disimpan akan hilang. Oleh karena itu RAM digunakan oleh sistem mikroprosesor untuk tempat penyimpanan sementara dari program dan data atau variabel yang sifatnya sementara. Dinamakan Random Access Memory karena lokasi maupun data dapat dicapai secara langsung dengan menempatkan address yang tepat pada inputnya.

Pada RAM terdapat parameter Read Access Time yaitu waktu yang diperlukan oleh data untuk menjadi stabil pada output pin pada memory setelah menerima permintaan dari mikroprosesor. Read Access Time ini merupakan parameter yang sangat penting karena menentukan kecepatan dari RAM (lihat gambar 2-9).

Parameter lain yang tak kalah pentingnya adalah Write Access Time yaitu waktu yang diperlukan sesudah address bus yang diberikan pada input stabil sebelum pulsa clock diberikan. Diagram waktu dapat dilihat pada gambar 2-10.

Ada 2 jenis RAM yang dapat digunakan oleh mikroprosesor untuk menyimpan data, yaitu :

- a. Static RAM
- b. Dynamic RAM

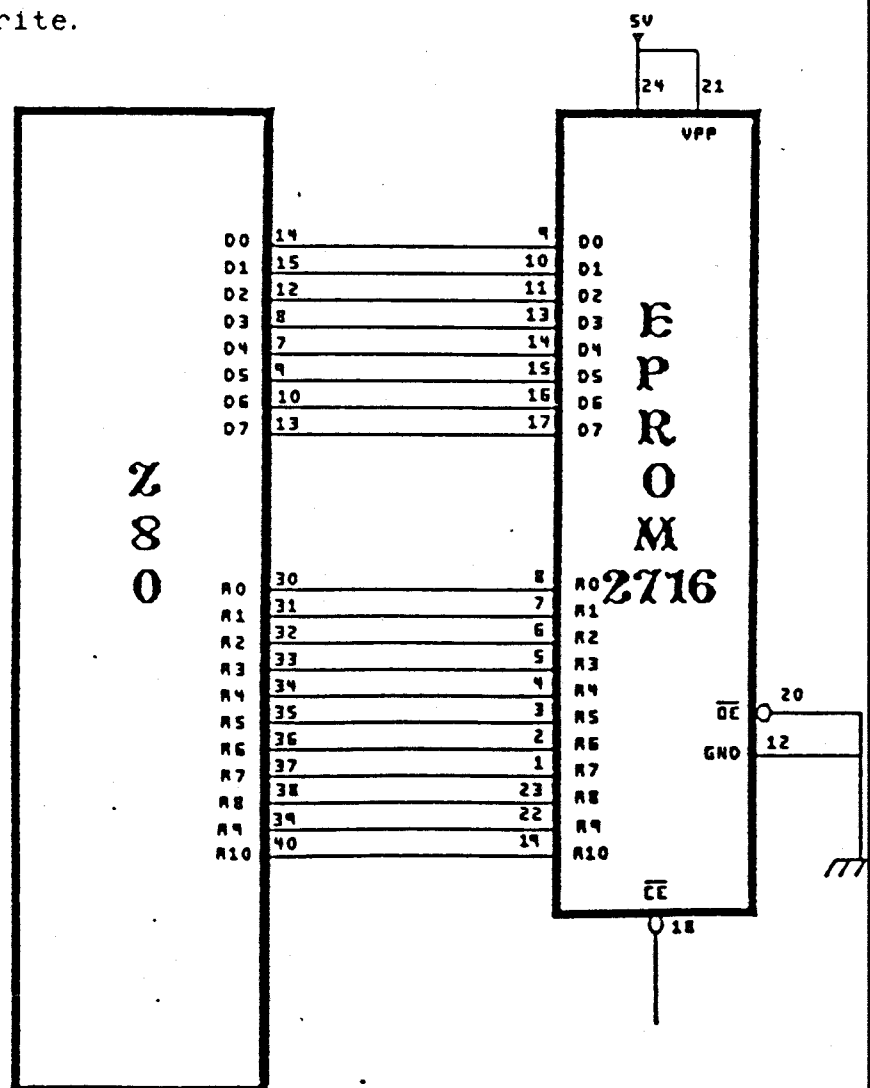
Static RAM dapat menyimpan data selama DC Power tidak diputuskan sedangkan Dynamic RAM harus diberi Refresh Cycle agar informasi dapat dipertahankan.

Perencanaan tugas akhir ini menggunakan Static RAM type 6116

yang berkapasitas 2048 x 8 bit sebagai tempat menyimpan data sementara.

### II. 1. 3. 2. 1. STATIC RAM

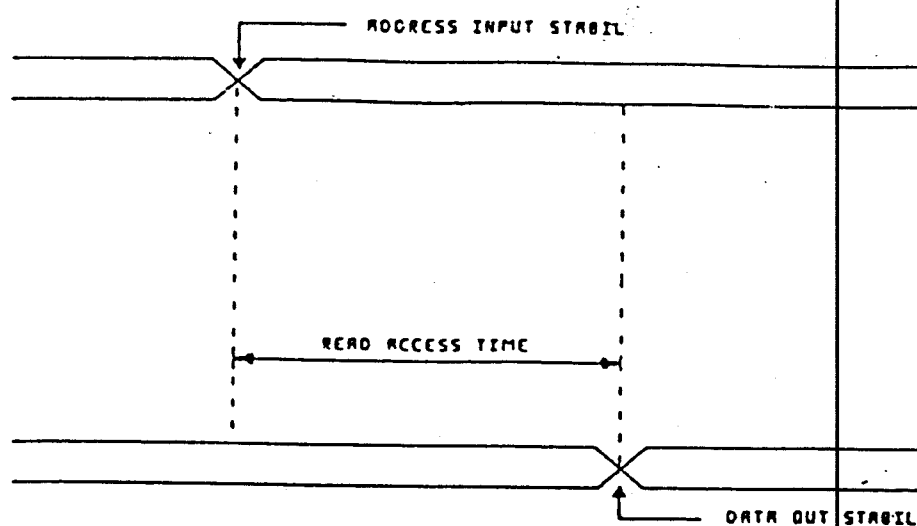
Static RAM adalah RAM yang dapat mengingat informasi yang disimpan di dalamnya tanpa perlu me-refresh informasi tersebut secara periodik. selama sumber daya listrik bekerja, semua informasi yang disimpan dalam Static RAM dapat diingat dengan baik. Jadi Static RAM hanya perlu dilakukan dua hal saja, yaitu Read atau Write.



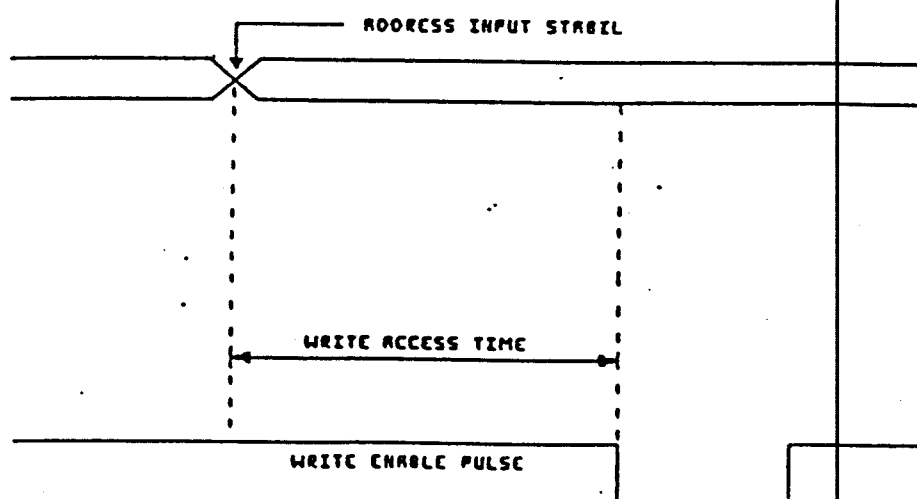
GAMBAR 2-6(3)

HUBUNGAN ANTARA Z-80 DENGAN EPROM 2716

3) ID14, Hal 1-200



GAMBAR 2-7(4)  
READ ACCESS TIME



GAMBAR 2-8(5)  
WRITE ACCESS TIME

4) Ibid , Hal 1-90

5) Ibid , Hal 1-210

#### II. 1. 3. 2. 2. DYNAMIC RAM

Perbedaan pokok antara Static RAM dan Dynamic RAM adalah diperlukannya sinyal Refresh pada Dynamic RAM secara periodik untuk mempertahankan informasi yang disimpan agar jangan hilang. Oleh karena itu pada Dynamic RAM selain Read and Write Cycle juga diperlukan suatu refreshing cycle.

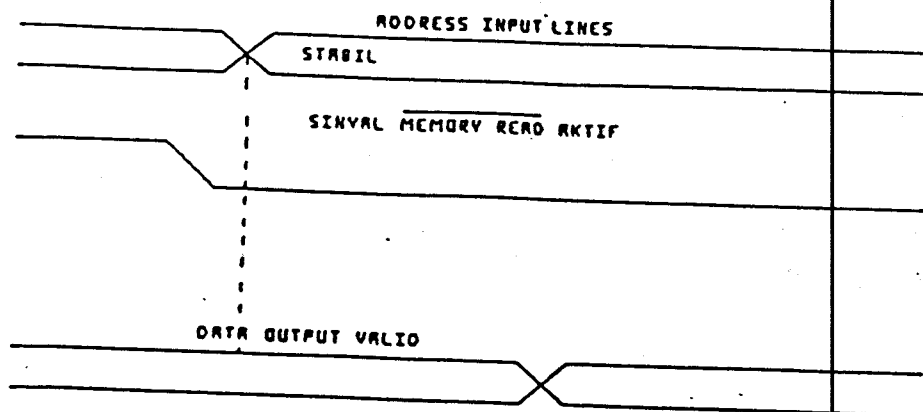
Dalam penggunaannya, Static RAM lebih praktis daripada Dynamic RAM karena tidak diperlukan suatu rangkaian Refresh. Namun demikian dari segi pembuatannya Dynamic RAM lebih mudah diimplementasikan, selain itu keuntungan yang lain adalah bentuk kemasan yang lebih kecil dan disipasi daya yang cukup rendah. Hal ini menyebabkan Dynamic RAM kadang - kadang dipertimbangkan untuk dipakai dalam suatu sistem mikroprosesor yang memerlukan memory dengan kapasitas yang besar.

#### II. 1. 3. 2. 3. MEMBACA DATA DARI RAM

1. Pertama - kali, memory menerima address yang menentukan lokasi tertentu. Rangkaian decoder dalam RAM tersebut memilih elemen mana yang diaktifkan.
2. Memory Read sinyal menjadi aktif dan memory langsung menerima sinyal ini .
3. Kemudian sistem menunggu dalam selang waktu tertentu (Read Access Time) sampai rangkaian di dalam RAM menjadi stabil .
4. Setelah Access Time, data akan muncul pada data bus dan dapat diterima oleh mikroprosesor. Jika mikro - prosesor terlalu cepat mengaktifkan sinyal Chip

Select (tidak menunggu Read Access Time), maka mikroprosesor akan mendapatkan data yang salah.

Diagram waktu pengambilan data dari RAM, seperti yang ditunjukkan pada gambar 2-11, mempunyai sistematika sbb :



GAMBAR 2-9

SINYAL MEMORY READ AKTIF

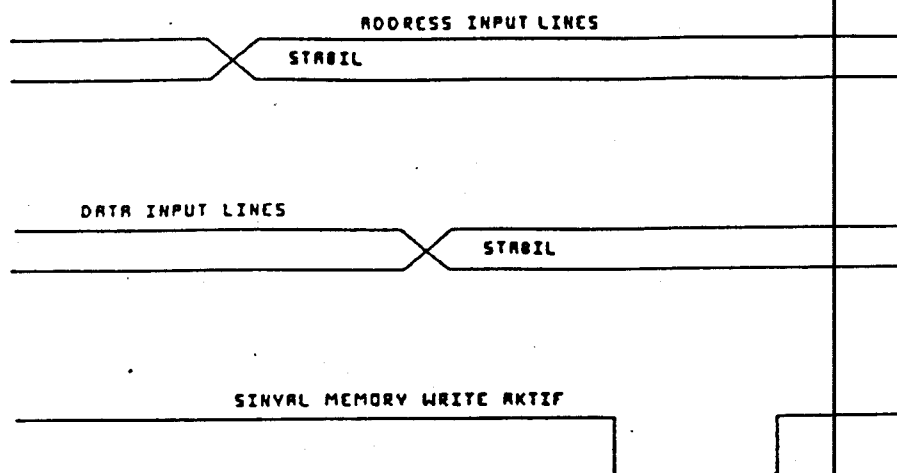
#### II. 1. 3. 2. 4. MENULIS DATA KE RAM

Penulisan data ke RAM urutannya dapat dijelaskan sebagai berikut :

1. Address dari data yang akan ditulis dimasukkan ke address dari memori yang sesuai dengan lokasi yang ditentukan.
2. Kemudian data yang akan ditulis ke memori diletakkan pada data bus .
3. Kemudian mikroprosesor menunggu untuk selang waktu tertentu (Write Access Time) untuk memberi kesempatan rangkaian decoding di dalam RAM stabil .
4. Setelah Write Access Time, maka sinyal memory write diaktifkan sehingga data yang terdapat pada input RAM ter -

tulis ke dalam elemen penyimpan RAM.

Gambar 2-12 menunjukkan diagram waktu penulisan data ke RAM.



GAMBAR 2-10

SINYAL MEMORY WRITE AKTIF

## II. 2. MIKROPROSESOR Z-80

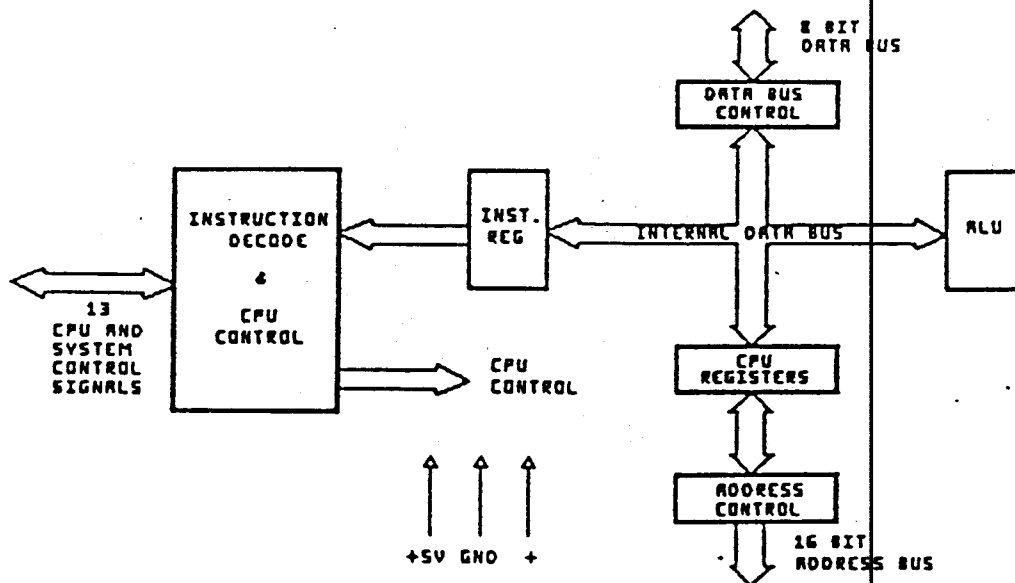
Mikroprosesor Z-80 ini dikemas dalam satu bentuk kemasan standard 40 pin Dual In Line (DIL). Mikroprosesor Z-80 ini dibuat oleh Multitech Industrial Corporation.

Keterangan untuk Arsitektur Internal mikroprosesor Z-80, khususnya mengenai Register - register serta flagnya secara lengkap, penjelasan secara rinci mengenai semua pin yang terdapat pada mikroprosesor, pengalamatan memori ( dari ROM hingga RAMnya ) maupun Input output ( Hubungan dengan rangkaian dekoder yang ditentukan berdasarkan timing diagramnya ), interkoneksi dengan rangkaian driver, Perpindahan data yang terdapat pada program yang terdapat pada RAM atau ROM menuju ke Mikroprosesor, semuanya akan dijelaskan pada Sub bab dari BAB utama mikroprosesor Z-80.



### II. 2. 1. ARSITEKTUR MIKROPROSESOR Z-80.

Blok diagram arsitektur internal dari Z-80 ditunjukkan pada gambar 2-11. Pada gambar tsb. ditunjukkan elemen - elemen penting yang dimiliki Z-80.



GAMBAR 2-11

#### BLOK DIAGRAM ARSITEKTUR INTERNAL Z-80

Internal register dari Z-80 ada 208 bit Read dan Write memori yang dapat diakses oleh programmer. Bit - bit ini dapat disusun dalam bentuk 18 buah 8 bit register dan 4 buah 16 bit register. Gambar 2-14 menunjukkan organisasi dari pasangan register Z-80. Semua register ini terbagi dalam 3 himpunan register, yaitu :

1. Accumulator (A) dan Flag (F)
2. Special Purpose Register
3. General Purpose Register

### 11. 2. 1. 1. ACCUMULATOR (A) DAN FLAG (F)

Register A (Accumulator) adalah register penampung Arithmetic Logical Unit, oleh karena itu seringkali disebut dengan singkat "accumulator". Accumulator ini menyimpan delapan bit hasil operasi aritmatika maupun logika. Accumulator biasanya menyimpan satu dari operand yang akan diolah oleh ALU untuk menjumlahkan isi dari register lain ke accumulator dan menyimpan hasilnya di accumulator itu sendiri untuk diolah lebih lanjut. Jadi accumulator merupakan tempat untuk menyimpan operand dari hasil - hasil operasi. Register F (Flag Register) merupakan register pembantu terhadap operasi aritmatik dan operasi logik. Bentuk bantuannya berupa menyimpan tanda keadaan bilamana terjadi carry, non carry, borrow, zero, parity/overflow. Letak definisi simpanan keadaan flag pada register F dapat dilukiskan sebagai berikut :

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-----							
S	Z	-	H	-	P/V	N	C
-----							

Bit - bit tersebut masing - masing bebas antara satu dengan lainnya, dan jika 0 disebut dalam keadaan reset, maka 1 disebut dalam keadaan set.

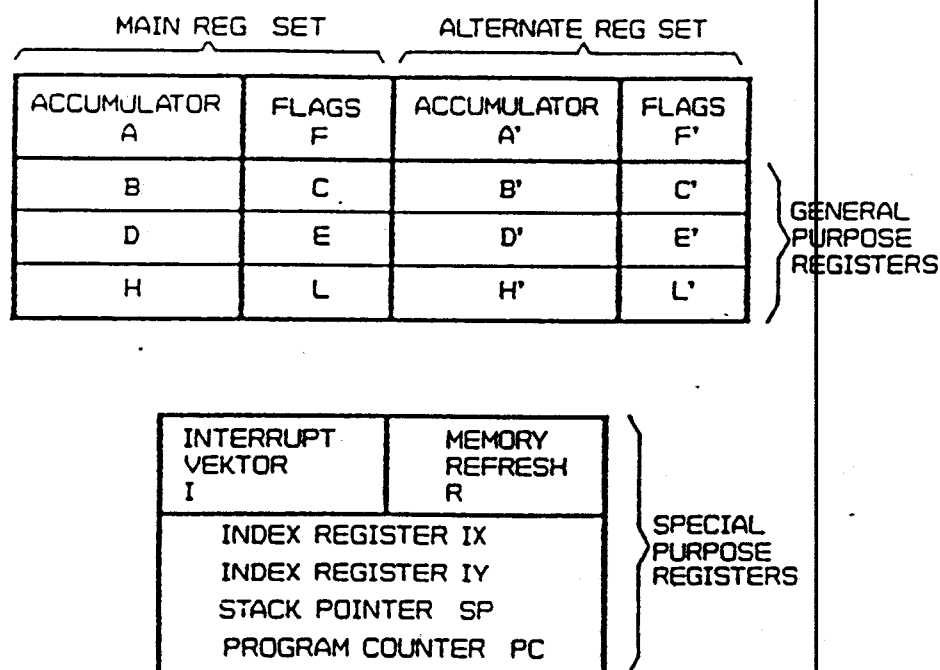
Bentuk kejadian dari register - register tersebut dapat ditabelkan sebagai berikut :

TABEL 2-2  
REGISTER FLAG

Bit	Nama Flag	0/1	Keterangan
0	C atau CY : carry	0	Operasi aritmatik atau logik tidak carry
		1	Operasi aritmatik atau logik terjadi carry
1	N : non carry	0	Operasi yang telah terjadi bukan SUBTRACT
		1	Operasi yang telah terjadi adalah SUBTRACT
2	P/V : parity/overflow	0	Parity even atau tidak terjadi overflow
		1	Parity odd atau terjadi overflow
3	Tidak digunakan		
4	H : half carry	0	Adisi/substraksi non carry di bit 4 register A
		1	Adisi/substraksi ada carry di bit 4 register A
5	Tidak digunakan		
6	Z : zero	0	Hasil operasi aritmatik atau logik tidak 0
		1	Hasil operasi aritmatik atau logik = 0
7	S : sign	0	Hasil operasi aritmatik atau logik = plus
		1	Hasil operasi aritmatik atau logik = minus

Accumulator dan flag ini mempunyai pasangan atau komplemen yang saling bebas dan berguna sebagai tempat cadangan bagi accumulator dan flag. Komplemen dari accumulator dan flag ini masing - masing dinotasikan dengan 'A' dan 'F'. Isi dari accumulator dapat ditukar dengan isi dari komplemennya. Contohnya bila accumulator diperlukan untuk melakukan suatu instruksi. tetapi isi dari

accumulator yang sekarang masih diperlukan untuk operasi lain pada instruksi berikutnya. supaya isi dari accumulator tidak hilang, isi dari accumulator ditukar dengan isi dari accumulator komplementnya. Dengan cara ini, isi accumulator semula tidak berubah dan dapat dipergunakan untuk operasi berikutnya.



GAMBAR 2-12

REGISTER - REGISTER PADA Z-80

### 11.2.1.2. SPECIAL PURPOSE REGISTER

Special purpose register ini terdiri dari lima macam register, yaitu :

#### 1. PROGRAM COUNTER

Program counter ini menyimpan 16 bit address dari instruksi yang dijemput (fetch) oleh mikroprosesor. setiap kali mengambil suatu instruksi maka isi PC dinaikkan dengan "1"

secara otomatis, sehingga PC akan menunjuk pada instruksi berikutnya. Apabila mikroprosesor menerima instruksi "JUMP" dari suatu subroutine maka program counter ini akan secara otomatis diisi dengan data address yang baru, di mana lokasi JUMP tersebut dituju.

## 2. STACK POINTER (SP)

Stack pointer adalah suatu register 16 bit yang berisi alamat lokasi teratas yang terisi pada stack. Cara penyimpanan dan pengambilan lokasi stack dilakukan dengan LIFO (Last In First Out), artinya bahwa data terakhir yang dimasukkan pada stack adalah data yang nantinya akan pertama kali keluar jika dipanggil. Instruksi untuk memasukkan data ke stack adalah "PUSH", sedangkan untuk mengambil data dari stack adalah "POP". Dengan adanya sistem stack ini dimungkinkan mikroprosesor untuk melakukan interrupt yang bertahap, melakukan nesting subroutine yang tak terbatas, penyederhanaan proses dan manipulasi data.

## 3. INDEX REGISTER IX DAN IV

Dua buah index register ini identik satu sama lainnya, menyimpan 16 bit address yang digunakan untuk mode indexed addressing. Pada mode ini, index register digunakan sebagai dasar (base) untuk menunjukkan lokasi memory dimana data dapat diambil dan disimpan. Tambahan byte yang terdapat pada instruksi yang diindexed (indexed instruction), menunjukkan banyaknya perpindahan yang dilakukan terhadap lokasi dasarnya (base location).

## 4. INTERRUPT PAGE ADDRESS REGISTER (I)

Mikroprosesor Z-80 dapat dioperasikan secara indirect mode (tidak langsung) pada lokasi di memori sebagai response dari adanya interrupt yang dikenakan pada mikroprosesor. Register I ini digunakan untuk menyimpan delapan bit orde tinggi indirect address tersebut. Untuk delapan bit address orde rendah, diberikan peralatan atau sistem yang meng-interrupt mikroprosesor.

#### 5. MEMORY REFRESH REGISTER (R)

Tujuh bit dari delapan bit register R ini secara otomatis menaik (increment) setelah melakukan instruksi fetch. Bit yang ke delapan akan tetap sesuai dengan yang diisikan dengan instruksi LD R,A. Register ini tidak diperlukan jika Z-80 menggunakan statik RAM.

### II. 2. 1. 3. GENERAL PURPOSE REGISTER

Ada dua kelompok general register ini, masing - masing kelompok ini saling bebas, terdiri dari enam buah delapan bit register, yang dapat digunakan sendiri - sendiri, atau sebagai tiga pasangan bit register. Kelompok ini dinotasikan sebagai BC', DE' dan HL'. Sama pada seperti accumulator, masing - masing isi register ini dapat ditukar dengan isi dari komplemennya. Register ini akan sangat membantu apabila terjadi operasi aritmatik dan logika.

### II. 2. 2. ARITHMETIC LOGIC UNIT (ALU)

Semua mikroprosesor mempunyai Arithmetic Logic Unit yang disebut ALU. ALU bertugas untuk melakukan operasi aritmatik dan logika dari data biner. Bekerjanya ALU ini selalu berhubungan

dengan register - register dan data bus. Adapun operasi - operasi yang dapat dilakukan oleh ALU adalah sebagai berikut : ADD, SUBTRACT, AND, OR, eXclusive OR (XOR), COMPARE, SHIFT AND ROTATE, INCREMENT, DECREMENT, SET BIT, RESET BIT dan TEST BIT. Hasil operasi mempengaruhi bit - bit flag.

### II. 2. 3. PIN - PIN PADA MIKROPROSESOR Z-80

Sifat dan fungsi dari pin - pin yang terdapat pada mikro-prosesor Z-80 ditunjukkan pada gambar 2-15 dan akan dijelaskan pada keterangan di bawah ini :

#### 1. A0 - A15 (address bus)

Merupakan bentuk output yang tri - state, terdiri dari 16 pin address, aktif jika berlogika "1" (aktif high), yang dapat mengamati lokasi memori sebanyak  $2^{16}$  atau 65536 lokasi dan untuk lokasi data I/O sampai dengan 256 lokasi input dan output. A0 adalah merupakan bit address yang paling rendah (least significant bit) dan A15 adalah bit address yang paling tinggi (most significant bit).

#### 2. D09 - D17 (data bus)

Merupakan bentuk input maupun output yang tristate, aktif pada logika "1", yang terdiri dari delapan pin data, yang digunakan untuk membawa sinyal data pada memori dan I/O.

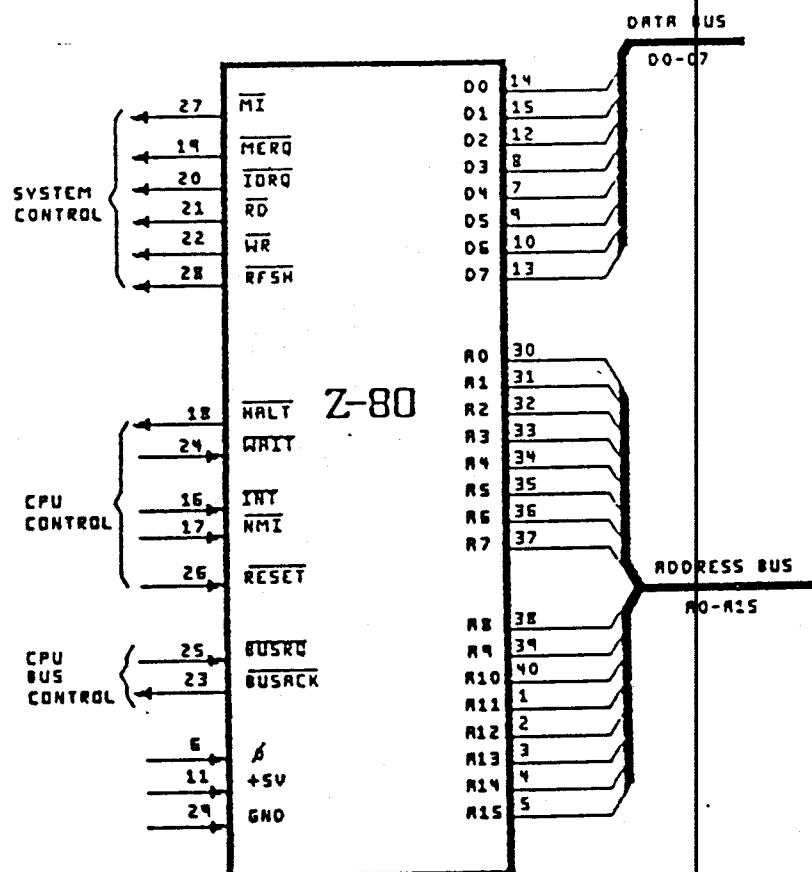
#### 3. M1 (machine cycle one)

Merupakan bentuk output, aktif bila berlogika "0" (active low). Sinyal M1 ini menunjukkan bahwa machine cycle yang sedang dilakukan adalah opcode fetch. Untuk dua byte opcode, sinyal M1 dibangkitkan untuk setiap opcode yang

difetch.

#### 4. MREQ (memory request)

Merupakan bentuk output yang tri-state, aktif bila berlogika "0" (active low). Sinyal MREQ ini menunjukkan bahwa address bus membawa sinyal address untuk operasi read



GAMBAR 2-13

#### KONFIGURASI PIN Z-80

#### 5. IORQ (I/O request)

Merupakan bentuk output yang tri-state, aktif pada logika "0" (active low). Sinyal IORQ ini menunjukkan bahwa address bus membawa sinyal address untuk operasi read atau pun write pada I/O.



#### 6. RD (read)

Merupakan bentuk output yang tri-state, aktif pada logika "0" (active low). Sinyal RD ini menunjukkan bahwa mikro-prosesor akan menerima data (read) dari memory atau I/O.

#### 7. WR (write)

Merupakan bentuk output yang tri-state, aktif pada logika "0" (active low). Sinyal WR ini menunjukkan bahwa mikro-prosesor akan mengirimkan data (write) pada memori atau I/O.

#### 8. RFSH (refresh)

Merupakan output yang aktif pada logika "0" (active low). Sinyal RFSH ini menunjukkan bahwa tujuh bit address orde rendah berisi refresh address untuk memori dinamik.

#### 9. HALT

Merupakan output, aktif pada logika "0" (active low). Sinyal HALT ini menunjukkan bahwa mikroprosesor sedang melakukan instruksi HALT, dan menunggu non-maskable ataupun maskable interrupt sebelum operasi berlanjut lagi. Pada saat HALT, mikroprosesor mengerjakan instruksi NOP untuk tetap mengaktifkan memory refresh.

#### 10. WAIT

Merupakan input, aktif pada logika "0" (active low). Sinyal WAIT berguna untuk memberi tanda kepada Z-80 bahwa memory atau I/O masih belum siap untuk mengirimkan atau menerima data. Mikroprosesor akan selalu menanti selama sinyal WAIT aktif. Dengan sinyal ini memungkinkan mikro-prosesor untuk bekerja dengan memory atau I/O secara selaras.

#### 11. INT (interrupt request)

Merupakan input, aktif pada logika "0" (active low). Sinyal interrupt request ini dihasilkan oleh peralatan I/O. Permintaan interrupt akan diterima pada akhir suatu instruksi jika :

- a. Interrupt enable flip - flop (IFF) telah diaktifkan dengan software
- b. Sinyal BUSRQ tidak aktif

Pada saat mikroprosesor menerima adanya interrupt, mikroprosesor mengeluarkan sinyal bahwa interrupt telah diterima pada awal instruksi berikutnya dengan sinyal IORQ dalam waktu M1.

#### 12. NMI (non maskable interrupt)

Merupakan bentuk input aktif bila ditrigger dengan pulsa dari logika "1" ke logika "0" (negative edge triggered). Interrupt NMI ini mempunyai derajat lebih tinggi dari pada interrupt lainnya (INT), artinya bila mikroprosesor sedang melakukan instruksi dari interrupt INT, kemudian ada sinyal interrupt NMI, maka instruksi dari interrupt akan ditinggal, dan mikroprosesor akan melakukan instruksi dari NMI. Adanya interrupt NMI ini mikroprosesor akan melakukan instruksi mulai pada lokasi 66 hexadecimal (0066H). Program counter akan disimpan di dalam Stack, sehingga program dapat kembali ke lokasi sebelum interrupt NMI dilakukan.

#### 13. RESET

Merupakan bentuk input, aktif pada logika "0" (active

low). Reset akan menyebabkan program counter menuju ke lokasi 0 hexadecimal (0000H) dan mikroprosesor akan diinisialisasi.

Inisialisasi mikroprosesor ini meliputi :

- a. Interrupt flip - flop akan dinonaktifkan
- b. Mengeset register I menjadi 00H
- c. Mengeset register R menjadi 00H
- d. Interrupt INT diset pada mode 0

Selama waktu RESET semua address bus dan data bus menjadi impedansi tinggi (high impedance).

### II. 3. UNIT INPUT OUTPUT

Unit input output merupakan inti di mana suatu sistem mikroprosesor dapat berkomunikasi dengan komponen - komponen di luar sistem tersebut. Unit input berfungsi untuk menerima informasi dari luar kemudian menterjemahkannya ke dalam bahasa sinyal digital yang dimengerti oleh mikroprosesor sehingga informasi dapat diolah dan diproses, sedangkan unit output berfungsi untuk menterjemahkan informasi yang telah diolah dan diproses tersebut ke dalam bahasa yang dimengerti oleh komponen di luar sistem tersebut.

#### II. 3. 1. PROGRAMABLE PERIPHERAL INTERFACE

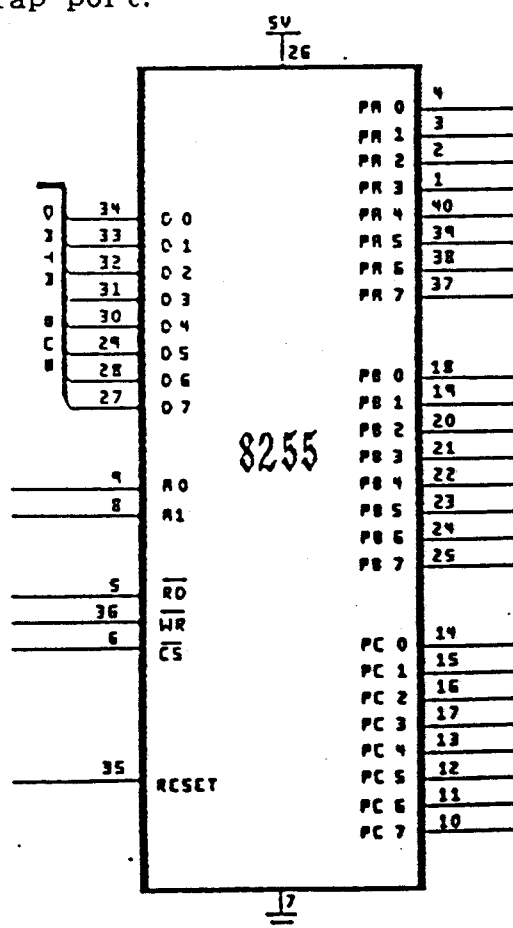
Programable Peripheral Interface 8255 dirancang sebagai interface dalam sistem mikroprosesor yang dikemas dalam bentuk 40 pin Dual In Line (DIP) yang termasuk jenis Large Scale Integration (LSI).

Gambar 2-14 menunjukkan diagram blok dari PPI 8255. Pada

gambar tersebut terlihat dua kelompok besar yang disebut kelompok kendali A dan kelompok kendali B. Kedua kelompok kendali tersebut mengendalikan empat kelompok I/O yang disebut :

- Port A ( PA0 - PA7 )
- Port B ( PB0 - PB7 )
- Port C Lower ( PC0 - PC3 )
- Port C Upper ( PC4 - PC7 )

Kelompok A mengontrol fungsi dari port A dan port C Upper, sedang kelompok B mengontrol fungsi dari port B dan port C Lower. Semua bagian dalam 8255 tersebut dihubungkan dengan internal data bus, dan melalui internal data bus inilah data dikirim atau diterima oleh setiap port.



GAMBAR 2-14

KOMFIGURASI PIN 8255

Port - port tersebut dapat digunakan dengan 3 mode, yaitu mode 0 (basic input/output), mode 1 (strobed input/output), dan mode 2 (bidirectional bus).

#### II. 3. 1. 1. KETERANGAN PIN PADA 8255

Fungsi dari masing - masing pin PPI 8255 dapat dijelaskan sebagai berikut :

1. Data Bus (DO - D7) :

Digunakan untuk input atau output dari peralatan luar, dimana semua informasi diterima dan dikirim melalui 6 bit data ini.

2. Chip Select (CS) :

Chip Select ini digunakan untuk mengaktifkan chip 8255, bila mendapat logika "0" mikroprosesor dapat mengirim data atau menerima data dari 8255.

3. Read (RD) :

Bila RD mendapat logika "0" dan CS juga mendapat logika "0", maka data output dari 8255 dapat dikeluarkan pada sistim data bus, dan siap dibaca oleh mikroprosesor.

4. Write (WR) :

Bila WR mendapat logika "0" dan CS juga mendapat 8255 melalui data bus.

5. Address Input (A0 - A1) :

Kombinasi dari kedua address input ini menentukan register mana dari 8255 yang akan menerima atau mengirim data dari atau ke mikroprosesor.

#### 6. Reset :

Fungsi dari pin ini adalah untuk mereset PPI 8255 dengan memberikan input logika "1". Pada saat reset ini semua I/O port diset dalam mode input.

#### 7. Port A (PA0 - PA7) :

Pin ini digunakan sebagai delapan bit input/output port untuk berhubungan dengan peralatan luar.

#### 8. Port B (PB0 - PB7) :

Pin ini fungsinya sama dengan port A, tetapi kedua port tersebut saling bebas satu dengan lainnya.

#### 9. Port C (PC0 - PC7) :

Pin ini fungsinya sama dengan port A maupun Port B, tapi pada port C ini dapat dibagi menjadi 2 kelompok yaitu Port C Upper (PC0 - PC3) dan Port C Lower (PC4 - PC7) masing - masing 4 bit yang digunakan untuk mengontrol peralatan luar yang berhubungan dengan 8255.

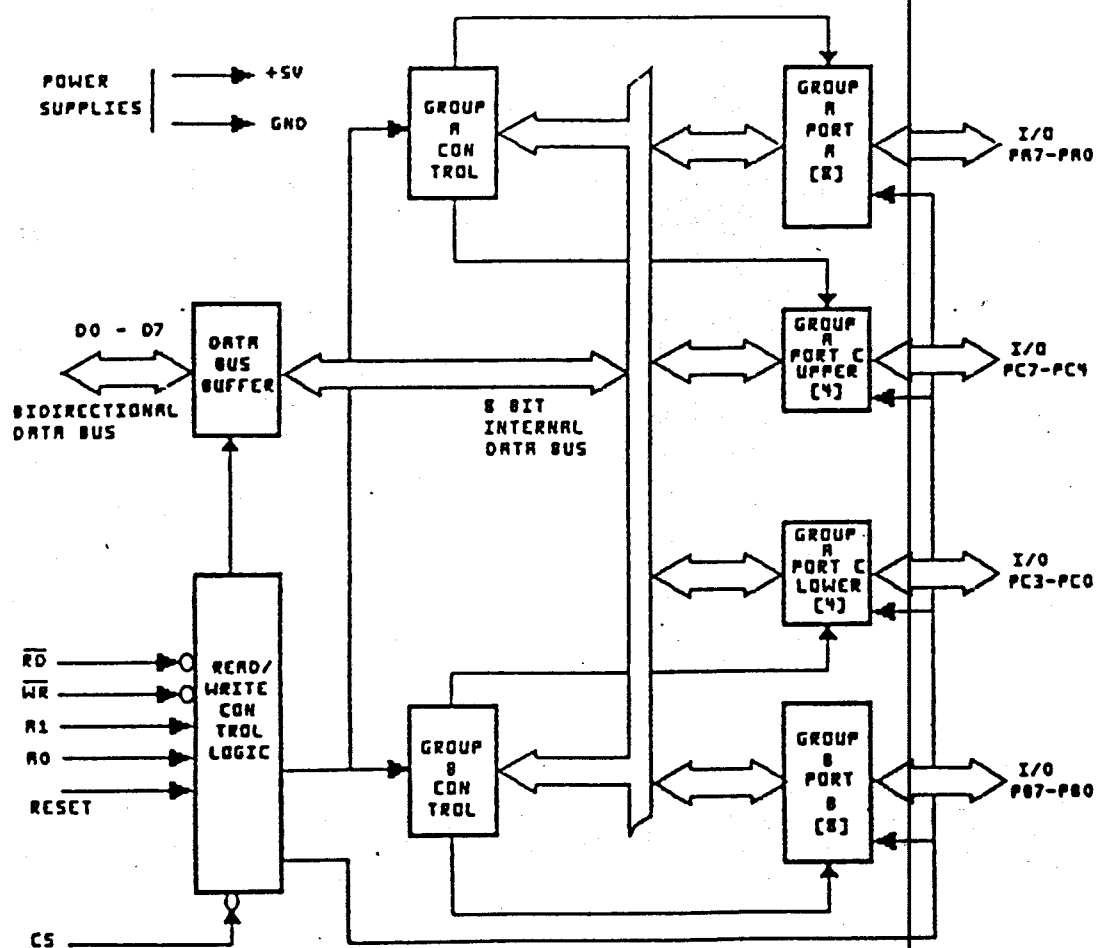
Konfigurasi dari pin - pin PPI 8255 dapat dilihat pada gambar 2-15.

### II. 3. 1. 2. CARA MENGHUBUNGKAN Z-80 DENGAN PPI 8255

Dua address input yang terdapat pada PPI 8255 yaitu A0 dan A1, bila dikombinasikan kedua logikanya akan diperoleh empat I/O address. Jalur alamat A0 - A1 dihubungkan langsung dengan A0 - A1 pada Z-80. Demikian pula dengan jalur data dari PPI 8255 langsung dihubungkan ke jalur data Z-80.

Untuk mengaktifkan PPI 8255 ini dengan jalan memberikan logika "0" pada kaki CS melalui dekoder. Sinyal RD dan WR pada PPI 8255 tidak dapat dihubungkan langsung dengan RD dan WR dari

mikroprosesor karena sinyal RD dan WR tersebut belum dipisahkan apakah sinyal tersebut untuk peralatan memory atau peralatan I/O. Untuk dipakai sebagai sinyal ke peralatan I/O maka dibutuhkan suatu rangkaian seperti pada gambar 2-18, sehingga output dari rangkaian ini bisa dihubungkan langsung ke 8255. Kegunaan dari rangkaian ini adalah menunjukkan bahwa sinyal RD dan WR benar - benar untuk peralatan I/O.



GAMBAR 2-15

## 8255 BLOK DIAGRAM

Pin RESET PPI 8255 tidak bisa dihubungkan langsung ke mikroprosesor Z-80, karena RESET pada Z-80 akan aktif pada logika "0", sedang pada 8255 akan aktif pada logika "1". Untuk itu maka pin RESET dari mikroprosesor Z-80 harus melalui sebuah inverter.

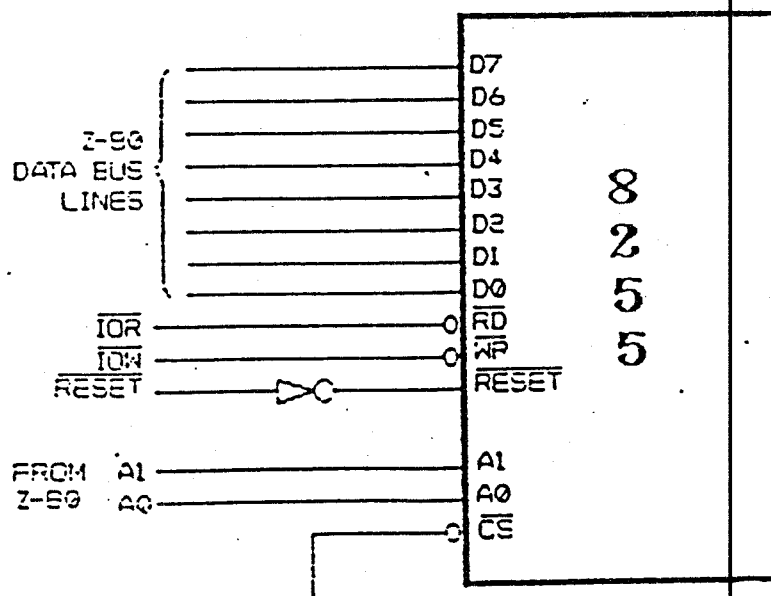
Gambar 2-17 merupakan rangkaian lengkap menghubungkan PPI 8255 dengan mikroprosesor Z-80.

### II. 3. 1. 3. REGISTER PADA 8255

Pada PPI 8255 mempunyai empat internal register, dengan mengatur register ini, maka 8255 dapat dioperasikan sebagai peralatan input atau output. Tabel 2-3 memperlihatkan hubungan antara sinyal WR, RD, serta address input A0 dan A1.

Fungsi dari register 0 (00B) sampai register 2 (10B), dapat dikirim melalui Control Word register 3 (11B). Fungsi dari control word ini adalah :

1. Menentukan fungsi setiap port
2. Menentukan mode yang digunakan



GAMBAR 2-17

HUBUNGAN ANTARA Z-80 DENGAN PPI 8255

### II. 3. 1. 4. PPI 8255 PADA MODE 0

Untuk mengaktifkan mode 0, mula - mula programmer harus



menulis control word pada control register. Control word inilah yang akan menentukan bagaimana register akan digunakan pada 8255. Penentuan bit pada control word untuk menentukan operating mode dapat dilihat pada Mode Definition Format gambar 2-20.

TABEL 2-3

## LOKASI DAN FUNGSI REGISTER PADA 8255

PIN 8255							NAMA REGISTER
RD	WR	A1	A0				
1	0	0	0				write port A
0	1	0	0				read port A
1	0	0	1				write port B
0	1	0	1				read port B
1	0	1	0				write port C
0	1	1	0				read port C
1	0	1	1				write control word
0	1	1	1				illegal read register!

Dengan mengaktifkan ketiga port sebagai output port, maka datanya harus demikian :

D0 D1 D2 D3 D4 D5 D6 D7

Control Word terdiri dari delapan bit data yang didefinisikan :

- D7 (mode set flag)

Menentukan aktif atau tidaknya 8255 pada saat sistem beroperasi. Aktif bila logik "1" dan tidak aktif bila logik "0".

- D6 dan D5 (mode selection group A)

Menentukan mode operasi group A. Mode 0 = 00, mode 1 = 01 dan mode 2 = 1X (X boleh 0 atau 1). Dalam tugas akhir ini menggunakan mode 0.

- D4 (port A)

Menentukan fungsi port A sebagai input atau output. Bila logik "1" berfungsi sebagai input dan bila logik "0" berfungsi sebagai output.

- D3 (port C Upper)

Menentukan port C Upper sebagai input atau output. Bila logik "1" berfungsi sebagai input dan bila logik "0" berfungsi sebagai output.

- D2 (mode selection group)

Menentukan mode operasi group B. Pada mode 0 logik "0".

- D1 (port B)

Menentukan fungsi port sebagai input atau output. Bila logik "1" berfungsi sebagai input dan bila logik "0" berfungsi sebagai output.

- D0 (port C Lower)

Menentukan fungsi 4 bit Port C Lower sebagai input atau output. Bila berfungsi sebagai input logik "0" dan bila berfungsi sebagai output logik "1".

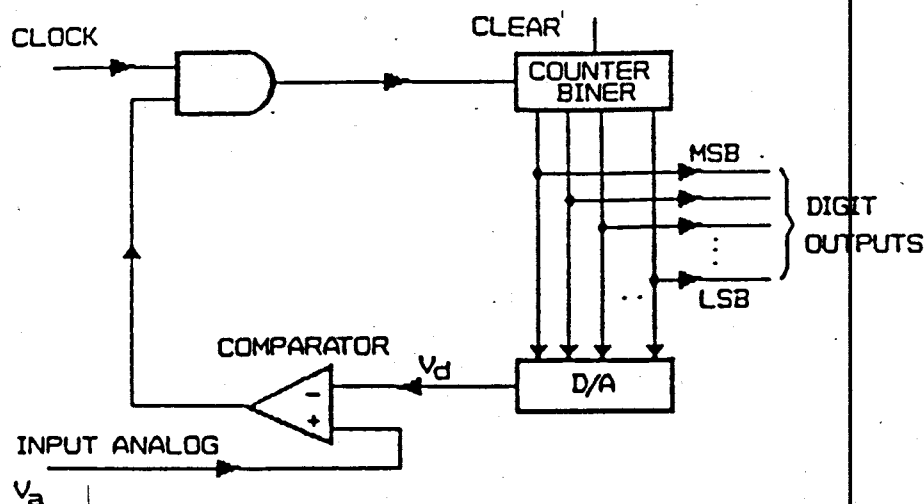
### II. 3. 2. ANALOG TO DIGITAL CONVERTER

Seringkali diperlukan agar data yang diambil dari suatu sistem fisik dalam bentuk besaran analog perlu dikonversikan dalam bentuk digital. Untuk itu dibutuhkan alat untuk mengubah informasi analog ke bentuk digital.

Dalam perencanaan tugas akhir ini digunakan ADC (Analog to

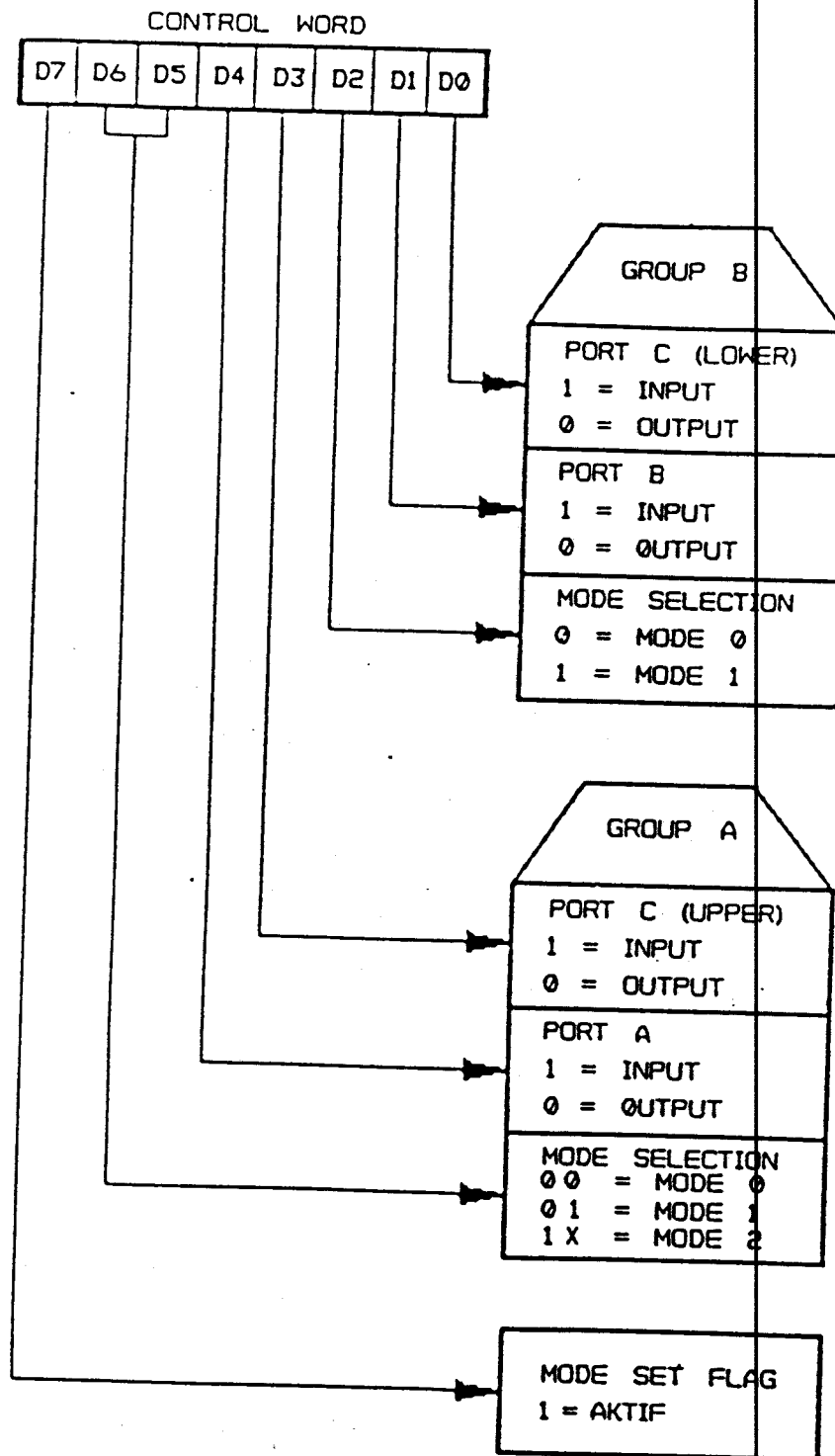
Digital onverter) 0808.

Pada gambar 2-21 ditunjukkan rangkaian Analog to Digital Converter yang menggunakan counter. Pulsa clear mengembalikan counter ke hitungan nol. Counter kemudian mencatat ke dalam bentuk biner jumlah pulsa dari saluran clock yang mempunyai selang waktu yang sama. Karena jumlah Clock merupakan Sumber pulsa yang dihitung sebanding dengan waktu, maka data biner yang mewakili hitungan ini digunakan sebagai input dari ADC yang outputnya merupakan gelombang bentuk tangga. Selama input  $V_s$  lebih besar dari  $V_d$  komparator, output komparator high dan gate AND terbuka untuk meneruskan pulsa clock ke counter. Jika  $V_d$  melebihi  $V_s$ , output komparator berubah ke low dan gate AND menjadi disable. Hal ini menghentikan hitungan pulsa pada saat  $V_s = V_d$  dan counter dapat dibaca harga digitalnya, yang menggambarkan tegangan input analog. Jika tegangan analog berubah - ubah terhadap waktu, maka tidak mungkin mengubah data analog terus menerus, akan tetapi bisa dilakukan dengan cara mensampel data analog tersebut pada interval tertentu.



Gambar 2-21

Pengubah ADC yang menggunakan Counter



GAMBAR 2-20

DEFENITION FORMAT 8255

# BAB III P E R E N C A N A N

## III. 1. BLOK DIAGRAM ALAT YANG DIRENCANAKAN

Alat yang digunakan dalam perencanaan tugas akhir ini menggunakan mikroprosesor Z-80 sebagai CPU nya. Adapun blok diagram secara keseluruhan dari sistem yang dibuat diperlihatkan pada gambar 3-1. Blok diagram ini terdiri dari beberapa bagian pokok yaitu :

### 1. Sistem Mikroprosesor

meliputi : Z-80, ROM, RAM, dekoder, PPI, display led.

### 2. Analog to Digital Converter

menggunakan ADC 0808 yang mengkonversikan tegangan analog yang disampel menjadi besaran digit yang akan diolah mikroprosesor.

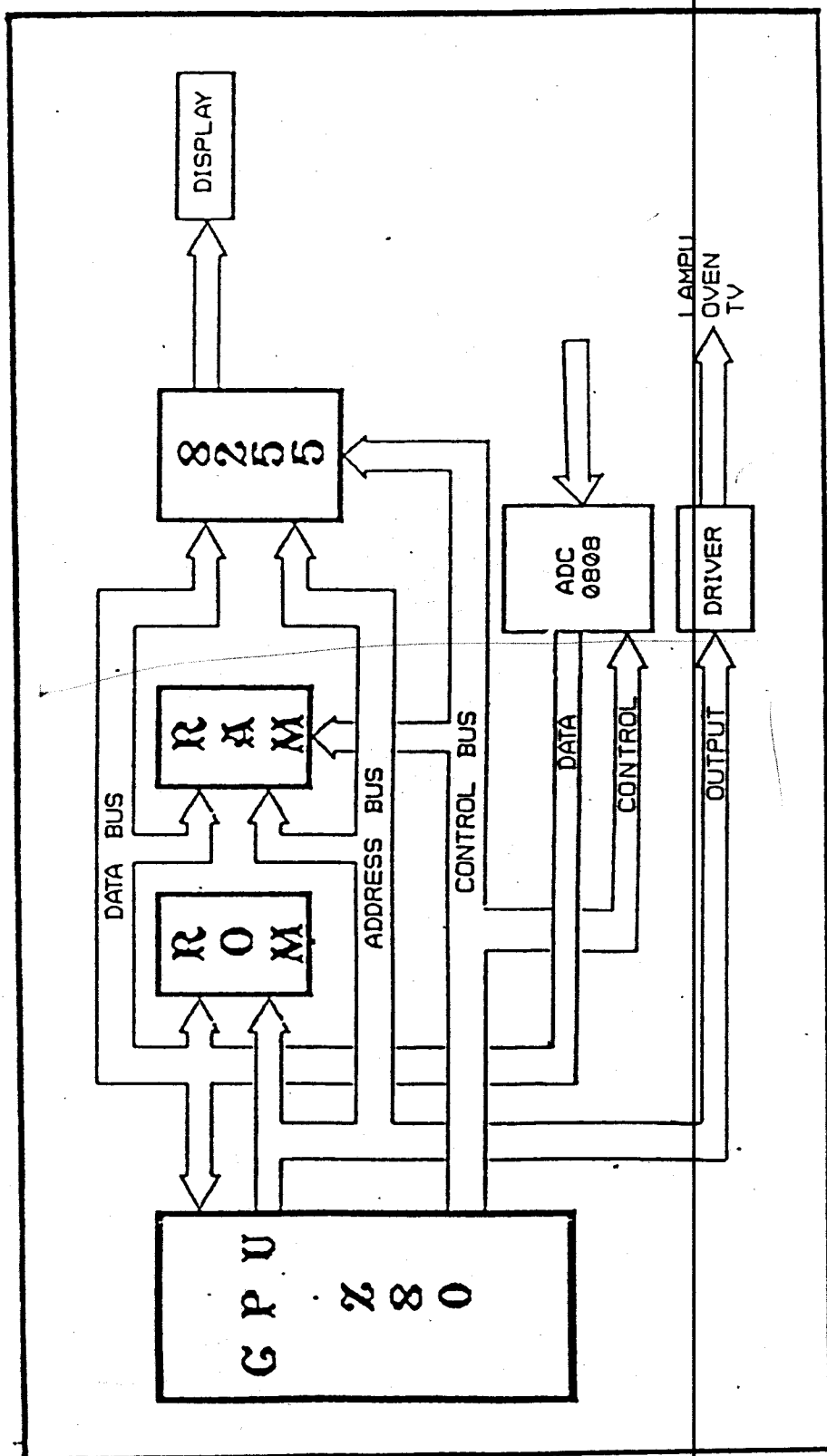
### 3. Driver

yang memungkinkan sinyal digital dari mikroprosesor untuk mengaktifkan port yang sesuai dengan harga yang dikonversi kan oleh ADC.

## III. 2. CARA KERJA ALAT

Adapun cara kerja alat ini adalah sebagai berikut :

Tegangan sampel yang dimasukkan akan diturunkan harganya oleh transformator , sehingga mendekati harga maksimum 5 volt untuk range 0-240 Volt AC. Tegangan input



GAMBAR 3-1

BLOK DIAGRAM DARI SISTEM YANG DIRENCANAKAN

tersebut dimasukkan ke ADC , lalu dikonversikan oleh ADC, harga konversi diolah oleh mikroprosesor , kemudian mikroprosesor akan mengirimkan sinyal ke port 8255 sesuai dengan , port yang dibandingkan oleh mikroprosesor .

### III. 3. SISTEM MIKROPROSESOR

#### III. 3. 1. RANGKAIAN OSILATOR

Rangkaian osilator berfungsi untuk membangkitkan pulsa (clock) yang diperlukan oleh mikroprosesor, yang terbentuk dari rangkaian RC, kristal, 2 buah inverter dari jenis IC TTL 74LS14 dan sebuah rangkaian D flip - flop dari jenis IC TTL 74LS74.

Mikroprosesor Z-80 dapat bekerja dengan clock 500 KHz sampai dengan 2,5 MHz (periode 400 sampai dengan 2000 nano detik).

Perencanaan alat ini menggunakan ROM dari jenis 2716 dan RAM jenis 6116 yang keduanya mempunyai access time yang sama, yaitu sekitar 100 sampai dengan 300 nano-detik. Jadi mikroprosesor dapat menerima atau mengirim data setelah menunggu waktu access time minimum selama 300 nano-detik.

Dalam lampiran B, Z-80 Characteristics dan AC Timing Diagram diperoleh :

Lebar pulsa MREQ pada saat low adalah :

$$Tw(MRL) = Tc - 30 \text{ nano detik}$$

Selang waktu dari MREQ low sampai data set - up adalah :

$$Tw(MRL) = TS (D) - Tdh(MR)$$

di mana :

$$Tw(MRL) = \text{lebar pulsa MREQ low}$$

$$= T_c - 30 \text{ ns}$$

$$T_s (D) = 50 \text{ ns}$$

$$T_{dh} (MR) = 100 \text{ ns}$$

Supaya data yang dikirim dan diterima ke memory absah maka  $T_w(MRL)$  harus lebih besar dari access time ROM dan RAM. Sehingga:

$$T_w(MRL) - 150 > 300 \text{ nano detik}$$

$$T_c - 30 - 150 > 300 \text{ nano detik}$$

$$T_c > 480 \text{ nano detik}$$

maka diperoleh periode clock  $< 2,083 \text{ MHz}$ .

Jadi periode clock yang diijinkan adalah :

$$500 \text{ KHz} < f_{\text{clock}} < 2,083 \text{ MHz}.$$

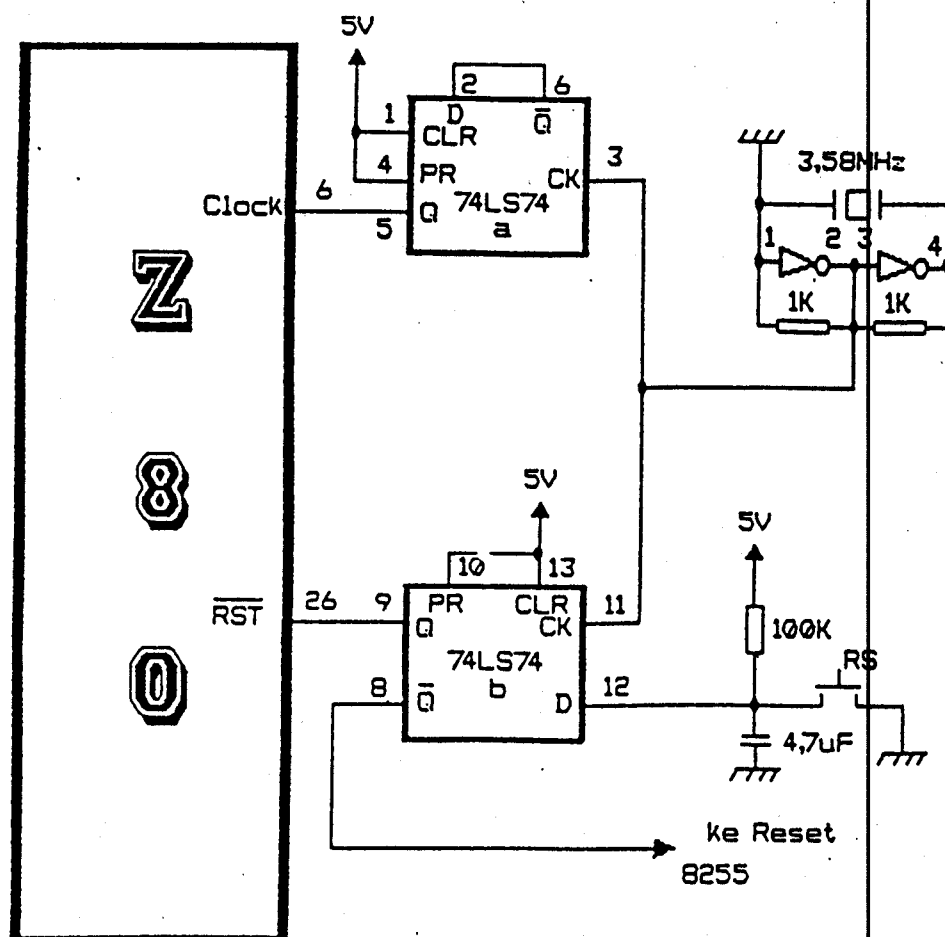
Kristal yang dipilih untuk perencanaan adalah sebesar  $3,579545 \text{ MHz}$  di mana kristal ini dimasukkan ke D flip - flop (74LS74) yang merupakan pembagi 2, sehingga diperoleh clock sebesar  $1,7897725 \text{ MHz}$  pada kaki 5 IC 74LS74 yang sudah dapat memenuhi persyaratan di atas.

### III. 3. 2. RANGKAIAN RESET

Rangkaian RESET berfungsi untuk me-reset semua kegiatan sistem mikroprosesor. Z-80 memerlukan sinyal low untuk me-reset dirinya. Pada saat sumber tegangan diberikan maka Z-80 akan diinisialisasi dan program mulai 0000H.

Selain dihubungkan ke pin 26 dari mikroprosesor, titik antara R dan C tersebut juga dihubungkan ke tombol RESET. Jika key tersebut ditekan, maka titik tersebut akan terhubung ke ground dan muatan pada kapasitor akan terbuang dengan cepat. Setelah key RESET tersebut dilepaskan, maka terjadilah proses inisialisasi seperti pada waktu keadaan power dihidupkan.





GAMBAR 3-2

## RANGKAIAN OSILATOR DAN RESET

III. 3. 3. RANGKAIAN DEKODER UNTUK MEMORI DAN I/OIII. 3. 3. 1. RANGKAIAN DEKODER UNTUK MEMORI

Rangkaian memori yang direncanakan menggunakan 2 buah chip 6116 dan 2 buah chip 2716. Chip 6116 adalah jenis memory static RAM yang berkapasitas 2 KByte sedangkan chip 2716 adalah jenis memory EPROM yang juga bisa berkapasitas 2 Kbyte. Ini berarti ada 5 zone bitnya. Sehingga address yang perlu dikodekan adalah : A15, A14, A13, A12 dan A11. Komponen yang digunakan untuk mengkode memory ini adalah IC jenis TTL 74LS138 yang merupakan 2 buah dekoder 2 bit.

Pembagian lokasi memori untuk RAM dan ROM ditentukan sesuai dengan peta memori yang ditunjukkan pada gambar 3-4. Untuk dapat mengamati ROM dan RAM maka ke 5 zona bit ini yang akan berperan untuk memilih lokasinya.

Lokasi ROM I tertinggi adalah pada alamat 07FFH yang dalam biner 0111 1111 1111B. Untuk lebih jelasnya lihat pada tabel 3-1 yaitu tabel kebenaran dari alamat memori. Dari tabel tersebut ternyata bahwa yang terjadi perubahan hanya pada bit alamat A11 dan A12.

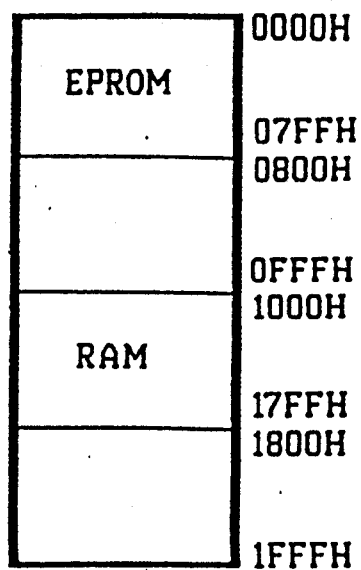
Untuk mengaktifkan IC dekoder 74LS138 ini maka pin 19 dari mikroprosesor Z-80 (MREQ) dihubungkan ke pin 1 dan 2 dari 74LS00 (NAND gate). Sedangkan keluarannya (pin 3) yang berlogika '1' dihubungkan ke pin 6 dari 74LS138 (input enable G1) yang aktif pada logika '1' dan bit alamat A14 dan A15 dihubungkan ke input enable G24 dan G25 yang aktif pada logika '0'.

Bit alamat A11, A12 dan A13 dihubungkan langsung ke input select A, B dan C. Dari ketiga input select ini akan diperoleh kombinasi keluaran Y0 sampai Y7. Untuk dapat mengamati alamat yang telah ditentukan sesuai dengan peta memori (gambar 3-4) maka keluaran Y0 dihubungkan ke EPROM I, keluaran Y2 dihubungkan ke RAM I dan keluaran Y3 dihubungkan ke RAM II.

Hal yang sangat penting yang harus diperhatikan dalam mengaktifkan IC 74LS138 ini adalah input enable G1 harus berlogika '1' dan input enable G2A dan G2B harus berlogika '0'.

### III. 3. 3. 2. RANGKAIAN DEKODER UNTUK I/O

Seperti halnya dengan dekoder memory, dekoder untuk I/O juga



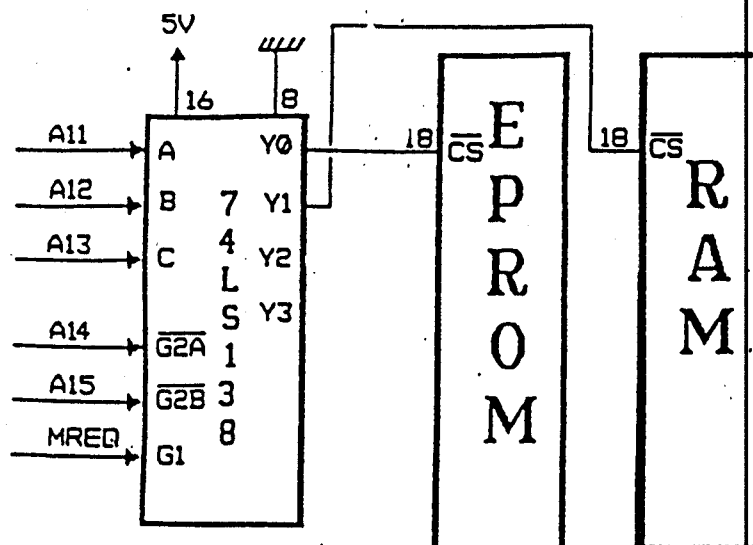
GAMBAR 3-3

PETA MEMORY

TABEL 3-1

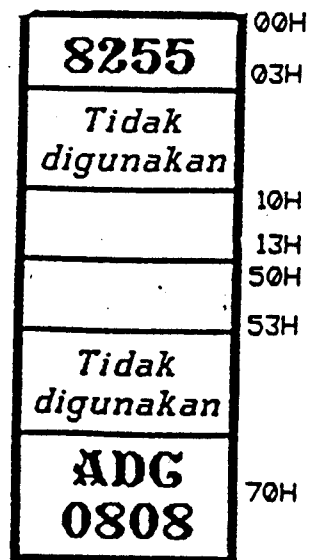
TABEL KEBENARAN DARI ALAMAT MEMORI

MEMORY	A15	A14	A13	A12	A11	A10	.	.	A1	A0	Alamat
EPROM	0	0	0	0	0	0	.	.	0	0	0000H
	...	0	0	0	0	1	.	.	1	1	07FFH
	0	0	0	0	1	0	.	.	0	0	0800H
	...	0	0	0	1	1	.	.	1	1	0FFFH
RAM	0	0	0	1	0	0	.	.	0	0	1000H
	...	0	0	1	0	1	.	.	1	1	17FFH
	0	0	0	1	1	0	.	.	0	0	1800H
	...	0	0	1	1	1	.	.	1	1	1FFFH



GAMBAR 3-4

RANGKAIAN DEKODER UNTUK MEMORI



GAMBAR 3-5

PETA PERALATAN I/O

menggunakan komponen yang sama yaitu 74LS138. Untuk dapat mengamati peralatan input & output digunakan bit alamat terendah yaitu A7 - A2. Dalam perencanaan ini yang digunakan untuk mengamati peralatan I/O hanya bit alamat A7 sampai A4.

Untuk mengaktifkan peralatan I/O sesuai dengan peta I/O (gambar 3-6), maka input enable G1 dari dekoder 74LS138 dihubungkan langsung ke supply 5V karena G1 aktif pada logika '1'. Sedangkan input enable G2A dan G2B dihubungkan ke bit alamat A7 dan IORQ yang aktif pada logika '0'. Input select A, B dan C dihubungkan ke bit alamat A6, A5 dan A4. Dengan kombinasi ini maka keluaran dari dekoder ini dapat digunakan untuk mengaktifkan peralatan I/O. Sesuai dengan peta I/O yang telah dirancang, maka untuk dapat mengaktifkan 8255, keluaran Y0, Y4 dan Y5 dari 74LS138 dihubungkan ke CS dari 8255.

Sedangkan untuk mengaktifkan ADC 0808 maka keluaran Y7 dari 74LS138 dan IORD serta IOWR dimasukkan ke IC 74LS02. dari kombinasi sinyal - sinyal ini dapat mengaktifkan ADC 0808 (untuk jelasnya lihat gambar 3-6).

#### III. 4. RANGKAIAN I/O

Input/output digunakan untuk menghubungkan sistem mikro-prosesor dengan dunia luar. Jenis LSI untuk input/output yang digunakan dalam perencanaan adalah PPI 8255, karena pengoperasiannya mudah. PPI 8255 juga mempunyai 3 buah port yang saling bebas. Ada tiga macam mode operasi 8255 yang dapat diprogram ke masing - masing port yakni :

- Mode 0 : Basic I/O
- Mode 1 : Strobed I/O
- Mode 2 : Bidirectional bus

Dalam perencanaan tugas akhir ini, PPI 8255 dioperasikan pada mode 0. Jadi di sini akan dibahas operasi mode 0 saja, sedangkan operasi mode yang lain dapat dilihat pada lampiran data 8255.

#### - MODE 0 (BASIC INPUT/OUTPUT)

Dalam mode 0, ketiga port 8255 berfungsi sebagai input atau output port biasa, artinya data dapat diambil/diletakkan dari/ke port secara langsung dengan instruksi IN/OUT ke port 8255 tanpa ada sinyal lain yang mengatur proses pengambilan/peletakan ini.

Semua port yang bekerja sebagai output port akan bersifat sebagai latch artinya data yang sudah diletakkan ke output port masih tetap ada selama tidak diubah oleh data lain yang ditulis ke port yang sama atau akibat adanya sinyal RESET. Hal ini memungkinkan bagi peralatan yang berhubungan ke output port untuk melakukan pembacaan yang tertunda artinya tidak pada saat data tersebut ada di output port. Tetapi bila port bekerja sebagai input port maka data yang akan dipindahkan dari peralatan input ke port 8255 harus sudah tersedia saat dilakukan pembacaan (tidak bersifat latch).

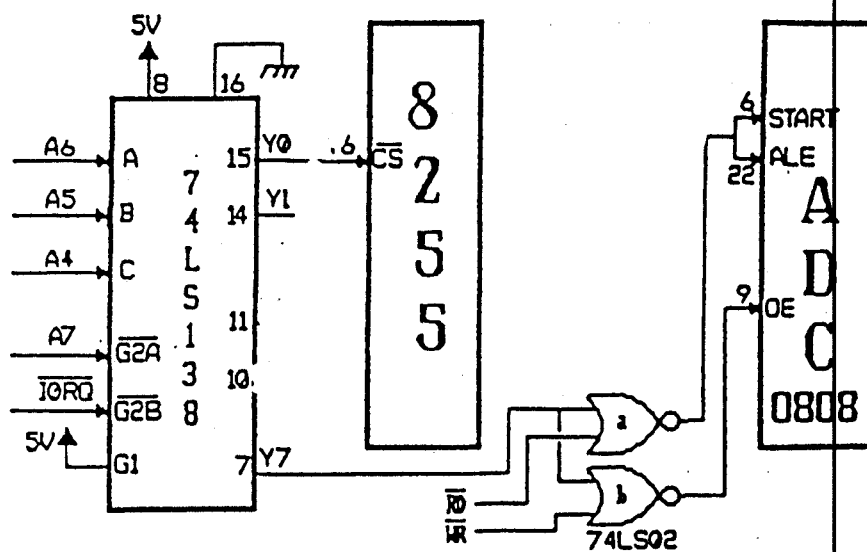
Dalam mode 0 tersebut ada 16 konfigurasi yang mungkin dibentuk yaitu kombinasi dari bit - bit D0, D1, D3 dan D4 dari control word register. Untuk operasi mode 0 pada semua port maka bit D7 = 1 dan bit D6, D5 dan D2 sama dengan nol. Keenam belas macam konfigurasi dari control word yang dapat dipakai terlihat pada tabel 3-3, beserta control word dalam hexadecimal yang harus

dipakai untuk tiap - tiap konfigurasi.

TABEL 3-2

TABEL KEBENARAN DARI ALAMAT I/O

I/O	A7 A6 A5 A4 A3 A2 A1 A0	Alamat
8255	0 0 0 0 0 0 0 0	00H
	0 0 0 0 0 0 1 1	03H
	0 0 0 1 0 0 0 0	
	0 0 0 1 0 0 1 1	
	0 1 0 1 0 0 0 0	
	0 1 0 1 0 0 1 1	
ADG	0 1 1 1 0 0 0 0	70H



GAMBAR 3-6

RANGKAIAN DEKODER UNTUK PERALATAN I/O

Dari ketiga port yang ada, satu port digunakan untuk input dan 2 port lainnya digunakan untuk output. Pembagian kerja dari ketiga port I/O ini adalah sebagai berikut :

1. Port A, sebagai input dengan pembagian fungsi :

a. PA0 digunakan sebagai input ADC (EOC) yang akan mengontrol sinyal dari ADC tersebut

2. Port B, sebagai output dengan pembagian fungsi :

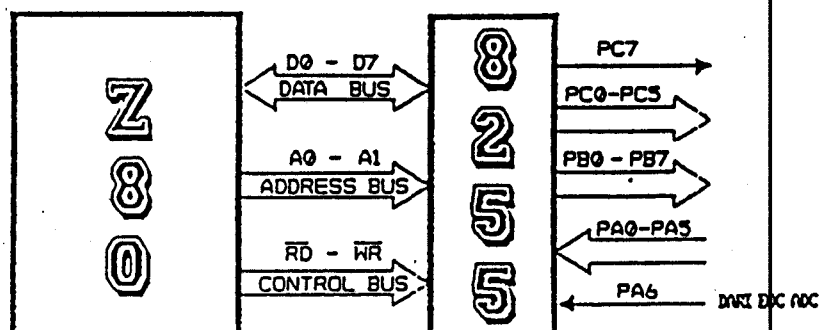
a. PB0 digunakan sebagai output ADC (START & ALE ) yang akan memberikan sinyal dari ADC tersebut .

b. PB1-PB4 digunakan sebagai output untuk mendriver device.

3. Port C, sebagai output dengan pembagian :

a. PC0-PC7 digunakan sebagai output untuk mendriver device.

Lokasi untuk PPI 8255 ini ditetapkan mulai pada alamat A0 - A3H. Gambar 3-7 adalah rangkaian I/O yang menggunakan PPI 8255.



GAMBAR 3-7

RANGKAIAN I/O YANG MENGGUNAKAN PPI 8255



### III. 5. RANGKAIAN ADC

Dalam perencanaan untuk sistem pengaturan tegangan listrik, maka digunakan chip IC ADC 0808. ADC ini mempunyai waktu konversi yang cukup singkat yaitu 100 ns, dan dapat di interface ke semua jenis mikroprosesor dengan penambahan tiap step 20 mv. Sebab itu penguatan pada bagian sensor dibuat 2 kali agar setiap 20 mV bisa direpresentasikan sebagai biner 1.

Untuk tegangan referensi ADC sebesar 5,12 V diperoleh dari konfigurasi zener dan buffer op-amp. Fungsi zener untuk memperoleh tegangan output yang tetap walau tegangan supply berubah - ubah atau impedansi input beban berubah - ubah.

Buffer digunakan agar diperoleh tegangan referensi yang impedansi outputnya kecil. Sinyal clock untuk ADC (pin 10) diperoleh dengan menggunakan IC 74LS04 yang dirangkai seperti pada gambar 3.10. Clock yang dihasilkan sekitar 250 KHz.

ADC 0808 mempunyai 8 input yang dimultiplex, di mana untuk pengontrolan input dilakukan oleh 3 pin kontroler yaitu pin 23, 24 dan 25. Cara pengontrolan ditunjukkan pada tabel 3-3.

TABEL 3-3  
KONTROL INPUT ADC

23	24	25	pin input
0	0	0	26 (IN0)
0	0	1	27 (IN1)
0	1	0	28 (IN2)
0	1	1	1 (IN3)
1	0	0	2 (IN4)
1	0	1	3 (IN5)
1	1	0	4 (IN6)
1	1	1	5 (IN7)

Karena dalam perencanaan ini hanya digunakan 1 buah input (IN0), maka ketiga pin kontroler digroundkan. Konversi dari sinyal analog ke digital dapat dirumuskan sebagai berikut :

$$\frac{V_{in}}{V_f - V_z} = \frac{D_x}{D_{max} - D_{min}}$$

di mana :

$V_{in}$  = tegangan input pada ADC 0808

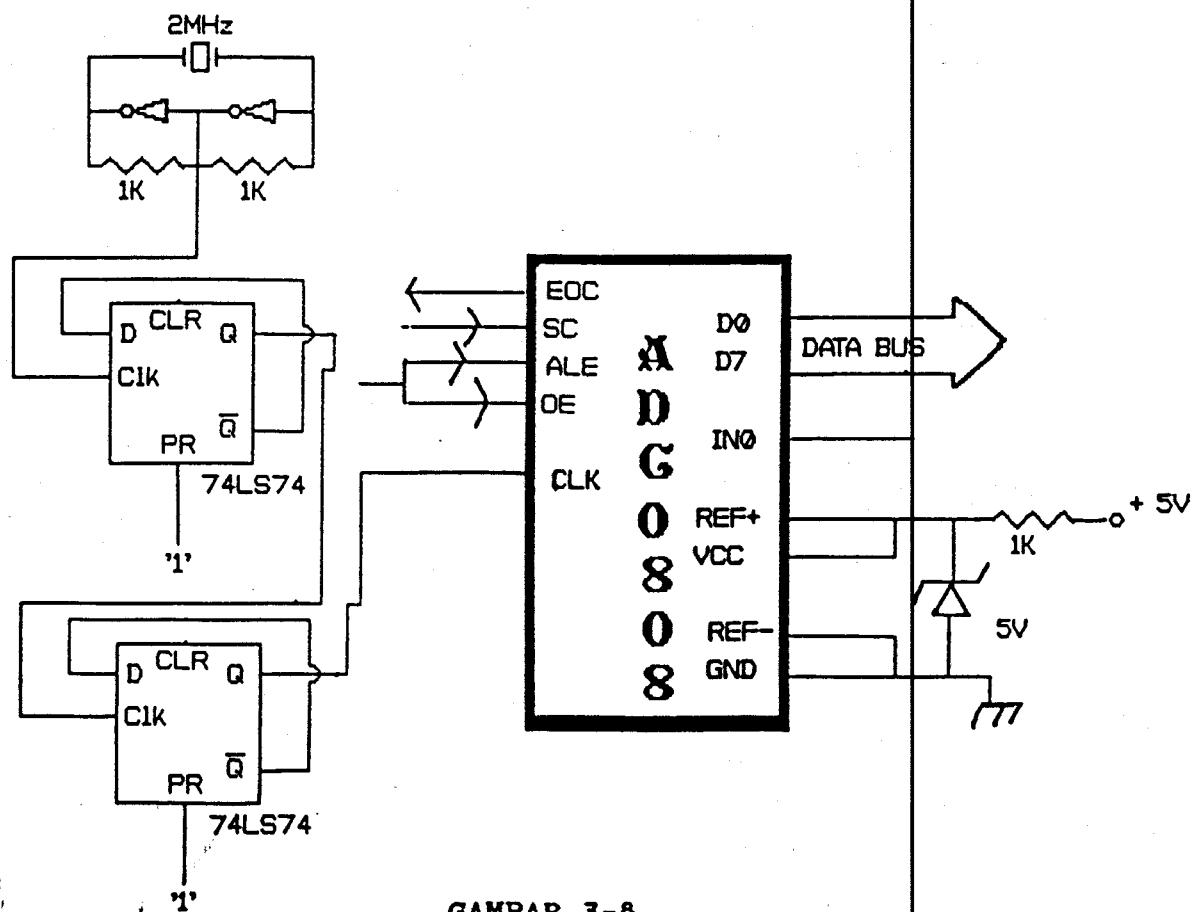
$V_f$  = tegangan referensi maksimum (pada skala penuh)

$V_z$  = tegangan nol

$D_x$  = data yang sedang diukur

$D_{min}$  = batas data minimum

$D_{max}$  = batas data maksimum



GAMBAR 3-8  
RANGKAIAN ADC 0808

Dari rumus di atas terlihat dengan jelas bahwa tegangan yang masuk pada salah satu terminal input ADC 0808 akan sebanding dengan data yang dikeluarkan. Karena ADC 0808 adalah A/D converter 8 bit maka data yang dihasilkan adalah  $2^8 = 256$  kemungkinan yang sama dengan input analognya.

Proses konversi dimulai dengan input pulsa ALE (Address Latch Enable) tinggi. Kemudian pemilih address dimasukkan ke dalam multiplexer dan gate - gate sinyal pada salah satu dari ke delapan channel input ke dalam komparator. Register successive approximation A/D konverter ini akan reset pada saat pulsa start conversion (SC) positive edge. Konversi akan dimulai pada saat pulsa start conversion menuju ke '0'. Suatu proses konversi akan di-interrupt dengan mengirimkan suatu pulsa start conversion yang baru. Proses konversi akan bisa berlangsung terus menerus dengan menghubungkan Keluaran End Of Conversion (EOC) ke masukan start conversion. Bila digunakan dalam mode ini, suatu pulsa start conversion external akan berlangsung setelah power up. Pulsa EOC akan menuju rendah antara 0 sampai 8 pulsa clock setelah sinyal rising edge dari start conversion. Sinyal rising edge dari EOC menunjukkan bahwa konversi telah selesai dan byte data dapat dibaca. Dengan memberikan pulsa aktif tinggi pada OE (Output Enable) maka data diletakkan pada ke 8 saluran data bus yang dalam keadaan normal tri state.

Instruksi untuk menuliskan ke beberapa port alamat, akan menyebabkan pulsa START dan ALE memulai konversi untuk channel yang dipilih.

Sinyal rising edge dari EOC menunjukkan bahwa data telah diperoleh, clock D flip - flop menset output Q dan meminta interrupt. Prosesor akan menjawab sinyal interrupt dengan memaksa EOC berikut.

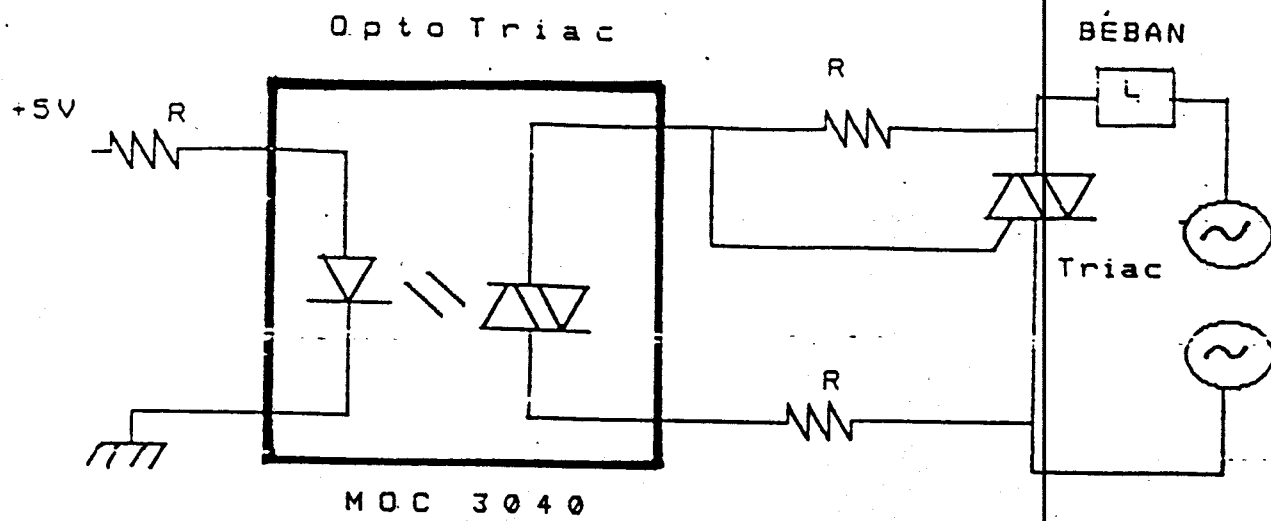
### III. 6. RANGKAIAN DRIVER

Untuk mendrive komputer, televisi, dan oven listrik, maka dibutuhkan suatu rangkaian yang dapat mengaktifkan peralatan - peralatan tersebut. Pada gambar 3-9, ditunjukkan suatu rangkaian driver yang dapat mendrive sebuah komputer. Komponen - komponen utama yang menunjang rangkaian ini adalah :

- a. Triac Q4004 L4
- b. opto TRIAC (MOC 3040)
- c. Dioda zener 5 V

Cara kerja rangkaian :

Saat input yang diberikan oleh perubahan tegangan jala - jala yang diturunkan oleh transformator 0.5 Ampere, dihubungkan dengan input ADC, maka input yang berupa data analog akan dikonversikan menjadi digital, kemudian data tersebut akan masuk kedalam suatu system yang diolah oleh mikroprosesor, data tersebut akan diolah oleh program untuk dibandingkan dengan tegangan yang sesuai dengan inputnya, setelah data berhasil diolah maka hasilnya akan dikirimkan ke port 8255, output port 8255 dihubungkan dengan driver yang mengaktifkan transformator utama yang akan memberikan output yang sesuai dengan inputnya.



GAMBAR 3-9  
RANGKAIAN DRIVER

# BAB II PEMBUATAN DAN PENGUKURAN ALAT

## IV. 1. PEMBUATAN ALAT

Setelah melalui tahapan perencanaan, maka dicoba untuk merealisasikan hasil dari perencanaan tersebut dengan membuat perangkat keras dan perangkat lunaknya untuk membuktikan kebenaran dari perencanaan. Ada 2 modul yang dibuat, yaitu modul untuk minimum sistem Z-80 dan modul untuk pengontrolnya.

Pada modul minimum sistem Z-80 terdapat komponen - komponen utama, yaitu CPU Z-80, memory, PPI 8255 serta komponen penunjang lainnya yang sangat mendukung bekerjanya sistem. Sedangkan pada modul pengontrol terdapat komponen - komponen MOC 3040, ADC 0808 serta komponen driver tegangan untuk komputer serta untuk oven listrik, televisi dan lampu - lampu. Maksud dari pembuatan modul minimum sistem adalah untuk memudahkan pemeriksaan dan pengukuran apabila terjadi kerusakan.

Untuk mengoperasikan peralatan yang dibuat ini, maka dibutuhkan perangkat lunak untuk mengatur proses kerja sistem. Perangkat lunak yang digunakan adalah bahasa Assembly. Pemilihan ini dilakukan mengingat pengendaliannya lebih tinggi dari bahasa komputer yang lain dan memungkinkan penulisan program yang lebih cepat dengan menggunakan memory yang lebih kecil jumlahnya.

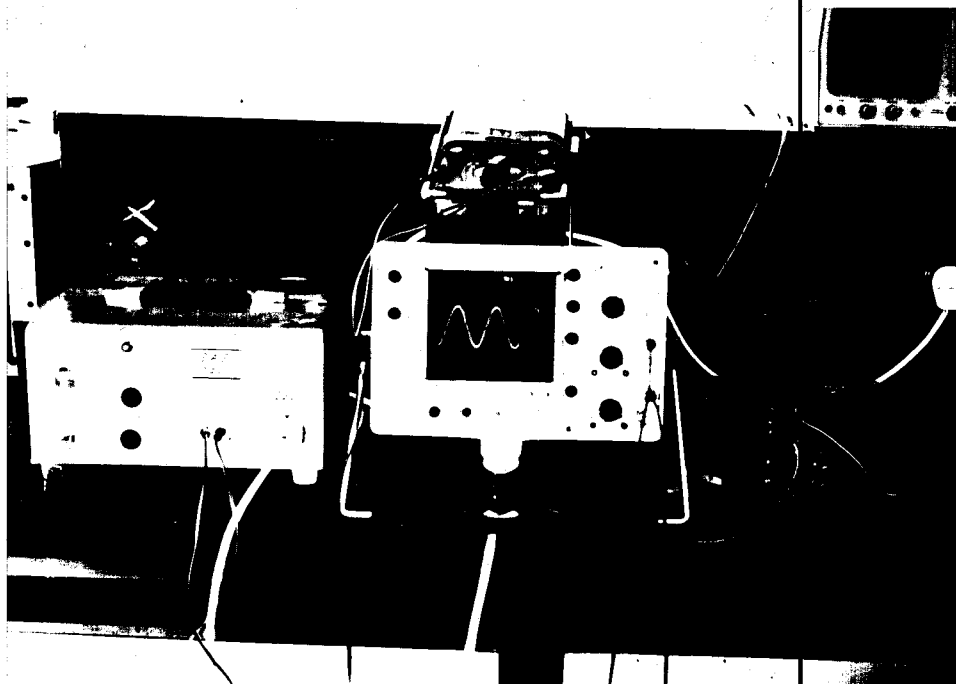
## IV. 2. PENGUKURAN ALAT

### IV. 2. 1. PENGUKURAN UNTUK FREKUENSI CLOCK Z-80

Dari perencanaan frekuensi clock yang diinginkan untuk Z-80 adalah sebesar 1,79 MHz. Pengukuran dilakukan dengan menggunakan oscilloscope. Hasil pengukuran pada pin 6 yang merupakan input clock Z-80 maka diperoleh hasil sebesar 1,789 MHz. Harga ini sangat mendekati frekuensi yang direncanakan.

### IV. 2. 2. PENGUKURAN FREKUENSI CLOCK ADC 0808

Dari perencanaan frekuensi input untuk ADC 0808 adalah sebesar 250 KHz. Frekuensi ini diperoleh dari rangkaian R dan C yang dihubungkan pada pin 6 IC 74LS14 diperoleh frekuensi sebesar 260 KHz. Ternyata ada perubahan sedikit dengan frekuensi clock yang diinginkan.



GAMBAR 4. 1

FOTO ALAT YANG DIBUAT

# IB A IB W IK IE SS II M IP U IL A N

---

Dengan telah berakhirnya penyusunan buku tugas akhir ini maka , akan diberikan sedikit hasil kesimpulan :

1. Untuk Pembuatan suatu sistem pengontrolan yang menggunakan mikroprosesor , maka hal yang paling penting , untuk diperhatikan adalah pemakaian setiap komponen perlu diketahui lebih dahulu spesifikasinya , agar dalam perancangan sistem tidak terjadi kesalahan .
2. Untuk membuat agar sistem dapat bekerja seefektif mungkin maka antara perancangan perangkat keras dan lunak harus bekerja saling menunjang sehingga dapat tercapai fleksibilitas yang tinggi dari suatu sistem .
3. Penggunaan mikroprosesor sebagai pengontrol tegangan listrik sangat praktis dan efisien .
4. Bagian yang terpenting dalam pembuatan sistem perangkat keras sistem adalah minimum sistem mikroprosesor , yaitu interkoneksi CPU , memori dan I/O . Sebab pada bagian inilah terletak pusat pengontrolan keseluruhan sistem.
5. Alat ini bekerja sesuai dengan apa yang semestinya diharapkan, waktu (timing ) , penyalan dan pemutusan tegangan diatur oleh program



# ID A I F T A I R I P U S T A I K A

1. Coffron, James W., Practical Hardware Details for 8080, 8085, Z80 and 6800 Microprocessors System. New Jersey : Prantice-Hall Inc., 1981.
2. Coffron, James W., Z-80 Assembly Language Programming, Sybex Inc., 1983.
3. Carr, Joseph J., Z-80 Users Manual, Virginia : Reston Publishing Company Inc., 1980.
4. General Electric, SCR Manual, New York : General Elctric Compa-ny., 1979.
5. Intel Corporation, Microsystems Components Handbook, Santa Clara : Intel Corporation., 1984.
6. Leventhal, Lance A., Z-80 Assembly Language Programming, Berke-ley : McGraw-Hill Inc., 1979.
7. National, Linear Data Book, California : National Semiconductor Corporation., 1982.
8. Spracklen, Kathe., Z-80 and 8080 Assembly Language Programming, New Jersey : Hayden Book Company Inc., 1979.
9. Texas Instruments., The TTL Data Book for Design Engineers, Dallas, Texas : Texas Instruments Inc., 1981.
10. Zaks, Rodnay., Microprocessor from Chip to Systems, Sybex, Inc., 1980.
11. Zaks, Rodnay., Programming The Z-80, Berkeley, California : Sybex, Inc., 1980.
12. Zilog Z80-Cpu, Z80A-CPU Technical Manual, California Zilog Inc., 1984.

; ---- PROGRAM UTAMA ----

ASEG

PORT00H EQU 00H  
 PORT01H EQU 01H  
 PORT02H EQU 02H  
 PORT03H EQU 03H  
 PORT07H EQU 07H

; --- INISIALISASI PORT ----

```

MULAI:          ORG 00H
                  LD B, 00H
                  DJNZ $
                  LD A, 90H
                  OUT (PORT03H), A
LOMPAT1:         XOR A
                  OUT (PORT01H), A
                  LD A, 0FFH
                  OUT (PORT01H), A
                  NOP
                  NOP
                  NOP
                  XOR A
                  OUT (PORT01H), A
  
```

; ----- MASUKKAN DATA DARI ADC -----

```

LOMPAT2:         IN A, (PORT00H)
                  AND 01H
                  CP 01H
                  JR NZ, LOMPAT2
                  IN A, (PORT07H)
                  LD D, A
                  SUB 0a5H
                  JR C, LOMPAT3
                  LD A, 80H
                  OUT (PORT02H), A
                  NOP
                  NOP
                  NOP
                  LD B, 0
                  DJNZ $
                  JR LOMPAT1
  
```

```

LOMPAT3:         LD A, D
                  SUB 0a3H
                  JR C, LOMPAT4
                  LD A, 40H
                  OUT (PORT02H), A
                  NOP
                  NOP
                  NOP
                  LD B, 0
  
```

DJNZ \$  
JR LOMPAT1

LOMPAT4: LD A, D  
SUB 0a1H  
JR C, LOMPAT5  
LD A, 20H  
OUT (PORT02H), A  
NOP  
NOP  
NOP  
LD B, 0

LOMPAT5: DJNZ \$  
JR LOMPAT1  
LD A, D  
SUB 098H  
JR C, LOMPAT6  
LD A, 10H  
OUT (PORT02H), A  
NOP  
NOP  
NOP  
LD B, 0

KEMBALI: DJNZ \$  
JR LOMPAT1  
JR LOMPAT1

LOMPAT6: LD A, D  
SUB 096H  
JR C, LOMPAT7  
LD A, 08H  
OUT (PORT02H), A  
NOP  
NOP  
NOP  
LD B, 0  
DJNZ \$  
JR LOMPAT1

LOMPAT7: LD A, D  
SUB 094H  
JR C, LOMPAT8  
LD A, 04H  
OUT (PORT02H), A  
NOP  
NOP  
NOP  
LD B, 0  
DJNZ \$  
JR KEMBALI

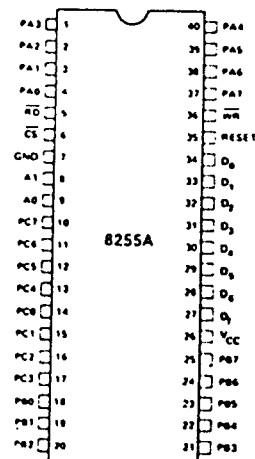
```
LOMPAT8:      LD A, D
               SUB 092H
               JR C, AKHIR
               LD A, 02H
               OUT (PORT02H), A
               NOP
               NOP
               NOP
               LD B, 0
               DJNZ $
               JR KEMBALI
AKHIR:        LD A, D
               SUB 090H
               JR C, KEMBALI
               LD A, 01H
               OUT (PORT02H), A
               NOP
               NOP
               NOP
               LD B, 0
               DJNZ $
               JR KEMBALI
; ----- SELESAI ----- SAS ' 90
```

## 8255A, 8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40 Pin Dual-In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in two groups of twelve and used in three major modes of operation. In the first mode (Mode 0), each group of twelve I/O pins may be programmed in sets of 4 to be input or output. In Mode 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining four pins three are used for handshaking and interrupt control signals. The third mode of operation (Mode 2) is a Bi-directional Bus mode which uses 8 lines for a bi-directional bus, and five lines, borrowing one from the other group, for handshaking.

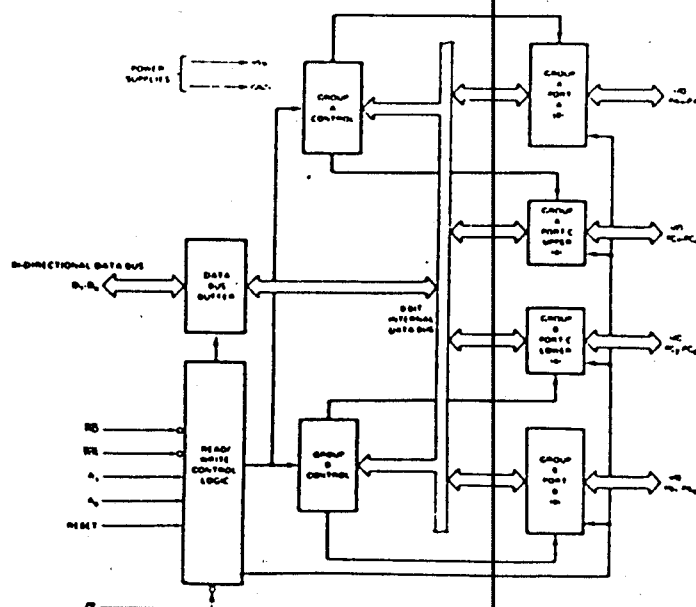
**PIN CONFIGURATION**



**PIN NAMES**

D <sub>7</sub> -D <sub>0</sub>	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
CS	CHIP SELECT
RD	READ INPUT
WR	WRITE INPUT
A <sub>0</sub> , A <sub>1</sub>	PORT ADDRESS
PA <sub>7</sub> -PA <sub>0</sub>	PORT A (8BIT)
PB <sub>7</sub> -PB <sub>0</sub>	PORT B (8BIT)
PC <sub>7</sub> -PC <sub>0</sub>	PORT C (8BIT)
V <sub>CC</sub>	+5 VOLTS
GND	0 VOLTS

**8255A BLOCK DIAGRAM**



### Input Control Signal Definition

#### STB (Strobe Input)

A "low" on this input loads data into the input latch.

#### IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

#### INTR (Interrupt Request)

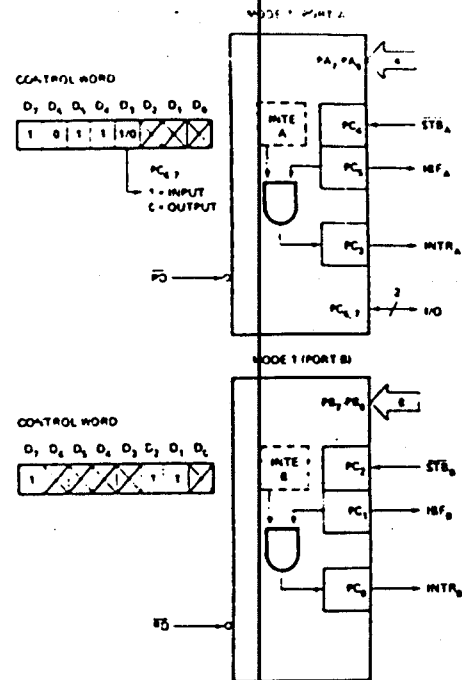
A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the STB is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

#### INTE A

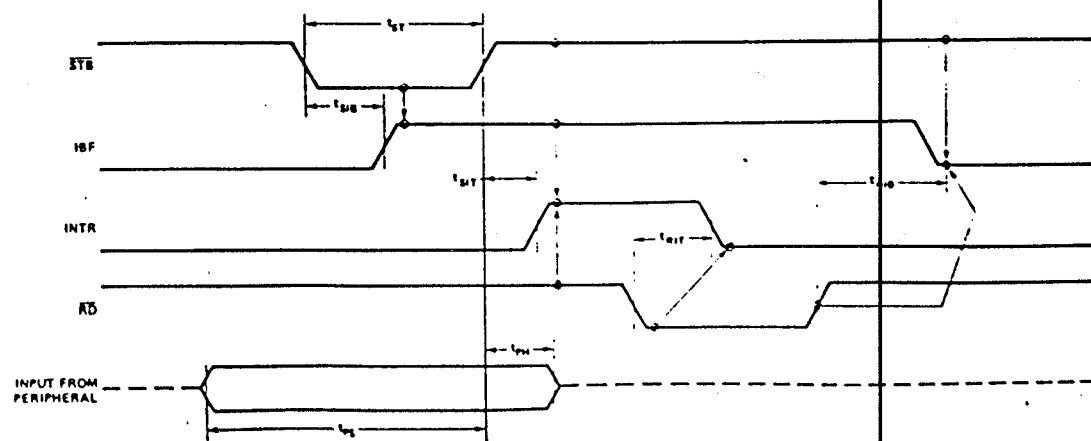
Controlled by bit set/reset of PC<sub>4</sub>.

#### INTE B

Controlled by bit set/reset of PC<sub>2</sub>.



Mode 1 Input



Mode 1 (Strobed Input)

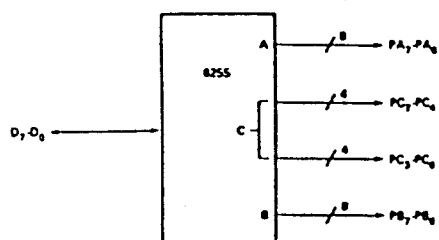
MODE 0 PORT DEFINITION CHART

A		B		GROUP A		GROUP B		
D <sub>4</sub>	D <sub>3</sub>	D <sub>1</sub>	D <sub>0</sub>	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 CONFIGURATIONS

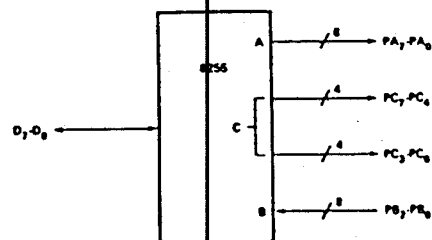
CONTROL WORD #0

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	0	0	0



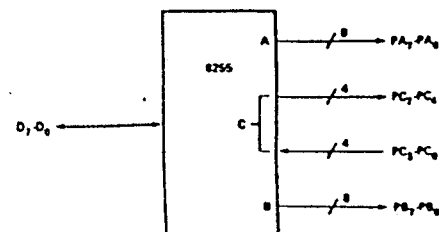
CONTROL WORD #2

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	0	1	0



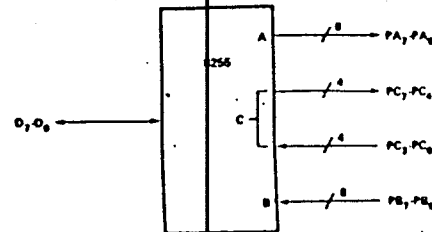
CONTROL WORD #1

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	0	0	1



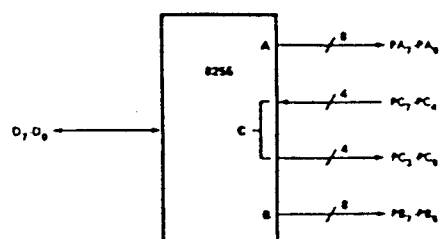
CONTROL WORD #3

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	0	1	1



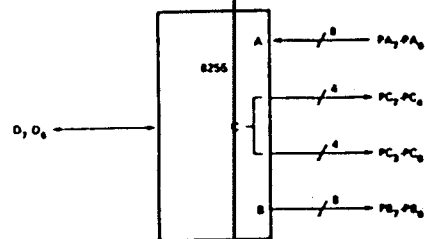
CONTROL WORD #4

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	0	1	0	0	0



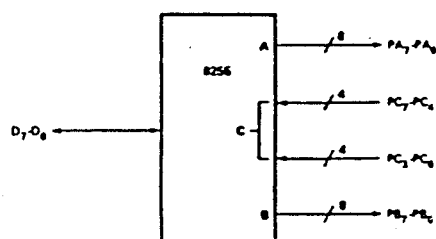
CONTROL WORD #5

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	1	0	0	0	0



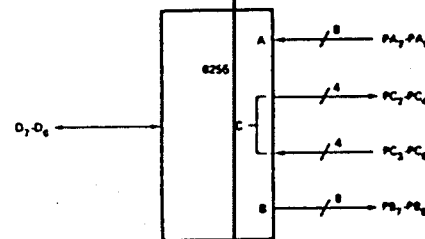
CONTROL WORD #6

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	0	1	0	0	1



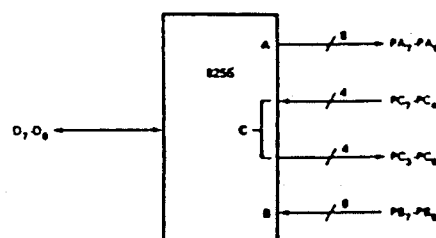
CONTROL WORD #9

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	1	0	0	0	1



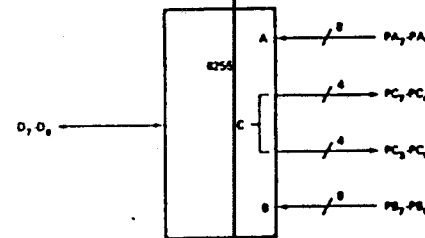
CONTROL WORD #8

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	0	1	0	1	0



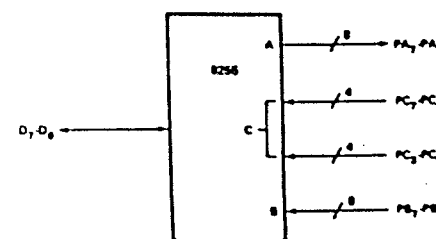
CONTROL WORD #10

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	1	0	0	1	0



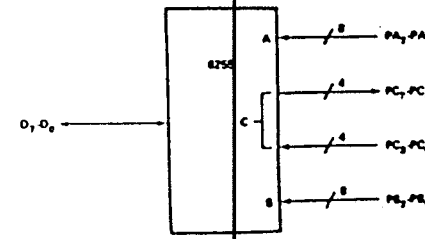
CONTROL WORD #7

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	0	1	0	1	1



CONTROL WORD #11

$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
1	0	0	1	0	0	1	1





## 8255 DETAILED OPERATIONAL DESCRIPTION

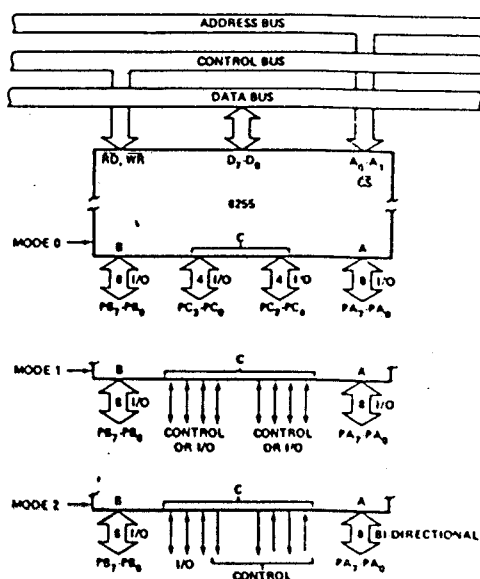
### Mode Selection

There are three basic modes of operation that can be selected by the system software:

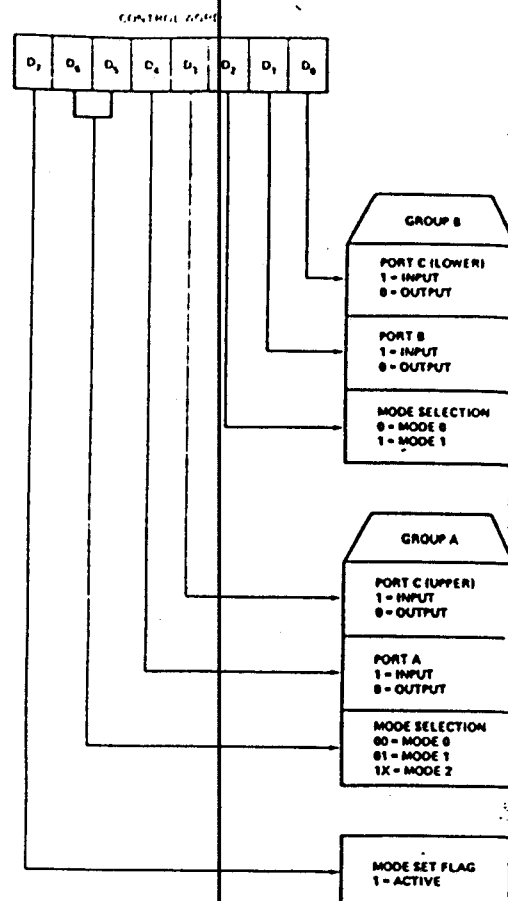
- Mode 0 – Basic Input/Output
- Mode 1 – Strobed Input/Output
- Mode 2 – Bi-Directional Bus

When the RESET input goes "high" all ports will be set to the Input mode (i.e., all 24 lines will be in the high impedance state). After the RESET is removed the 8255 can remain in the Input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single OUTput instruction. This allows a single 8255 to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.



Basic Mode Definitions and Bus Interface

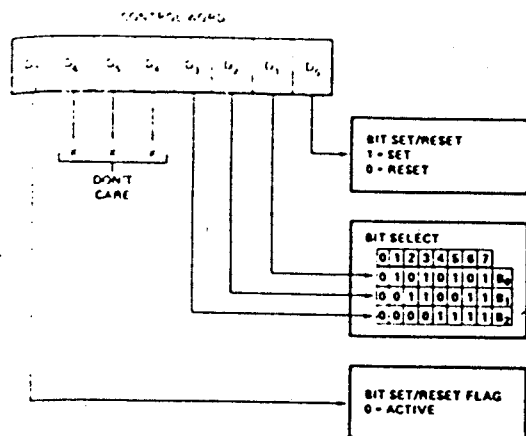


### Mode Definition Format

The Mode definitions and possible Mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255 has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

### Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.



Bit Set/Reset Format

## Operating Modes

## Mode 0 (Basic Input/Output)

This functional configuration provides simple Input and Output operations for each of the three ports. No "hand-shaking" is required, data is simply written to or read from a specified port.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

## Interrupt Control Functions

When the 8255 is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit set/reset function of Port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

## INTE flip-flop definition:

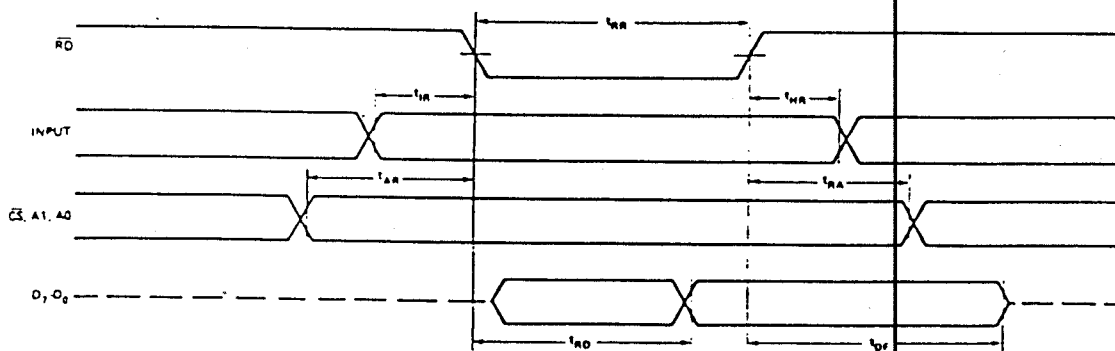
(BIT-SET) – INTE is SET – Interrupt enable

(BIT-RESET) – INTE is RESET – Interrupt disable

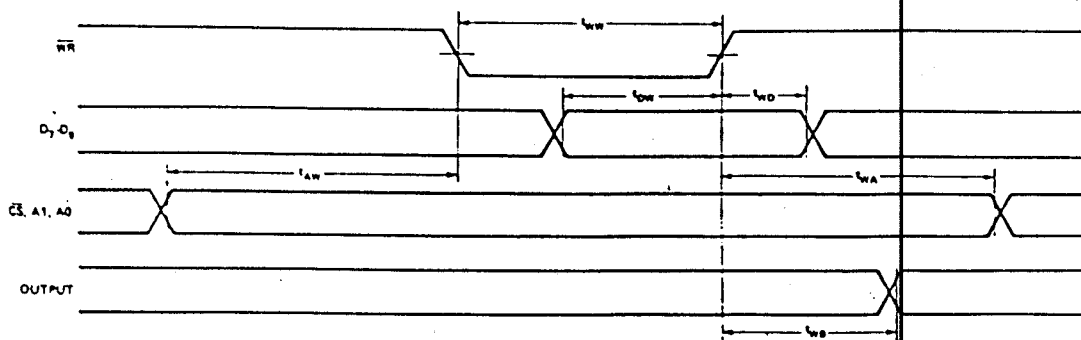
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

## Mode 0 Basic Functional Definitions:

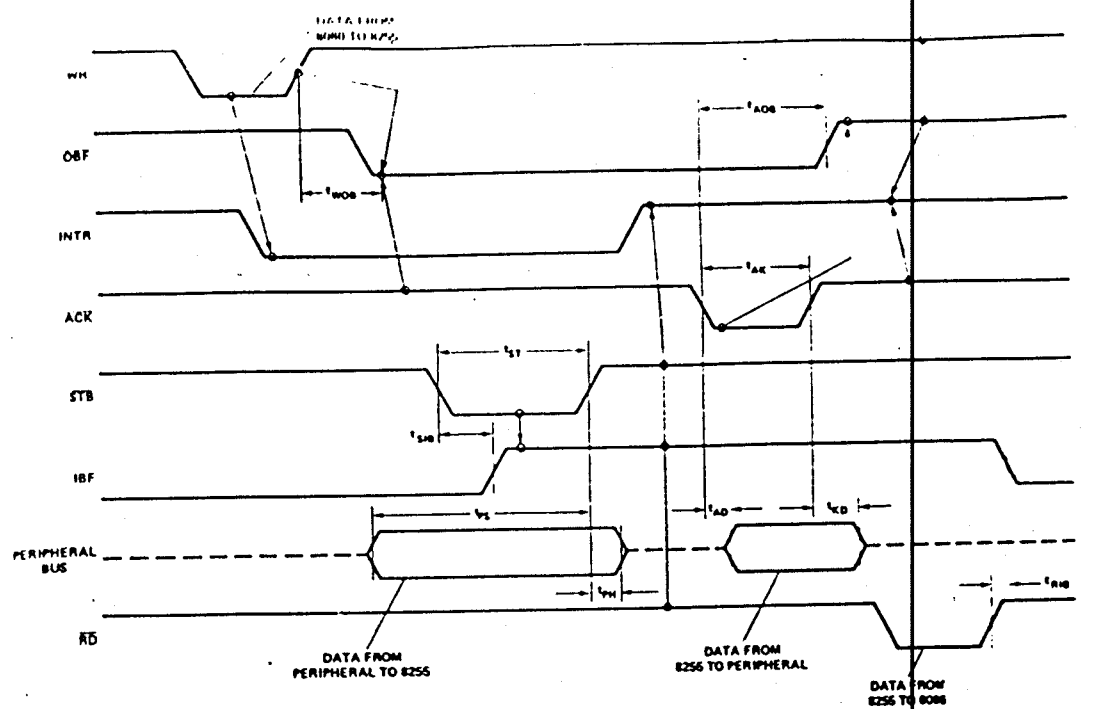
- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



Mode 0 (Basic Input)

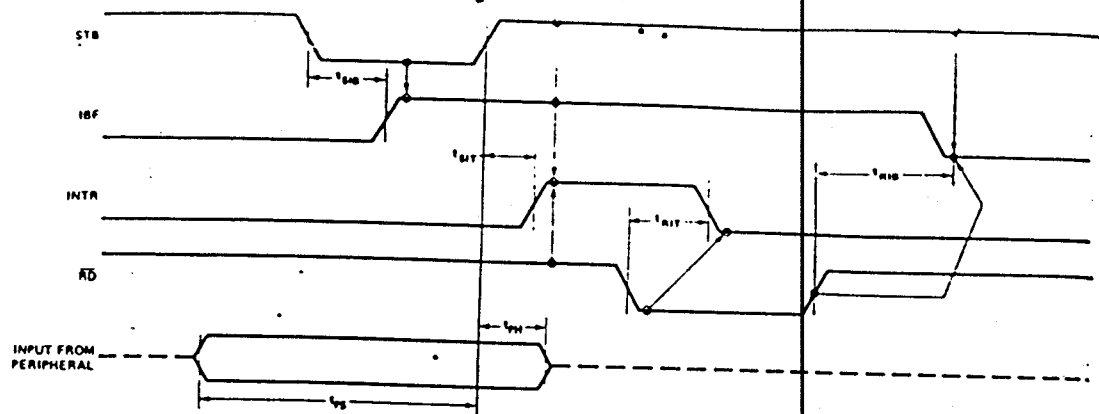


Mode 0 (Basic Output)

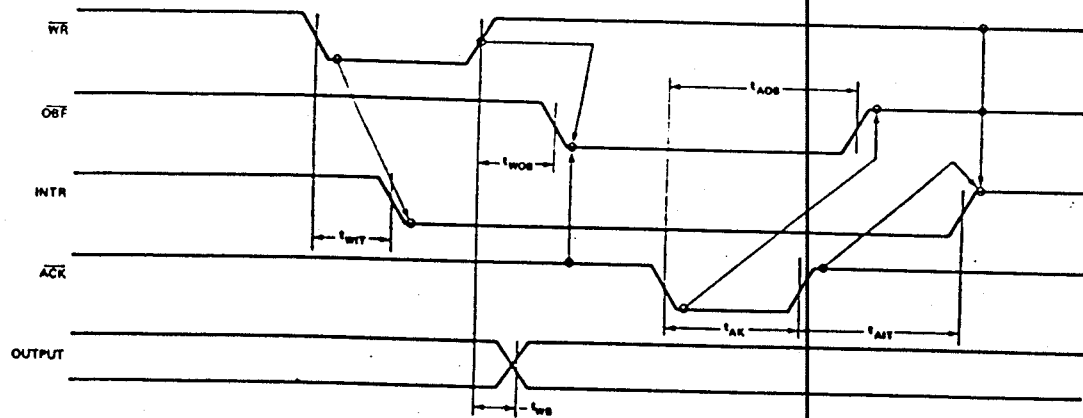


### Mode 2 (Bi-directional)

NOTE: Any sequence where  $\overline{WR}$  occurs before  $\overline{ACK}$  and  $\overline{STB}$  occurs before  $\overline{RD}$  is permissible.  
 $(INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + OBF \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR})$



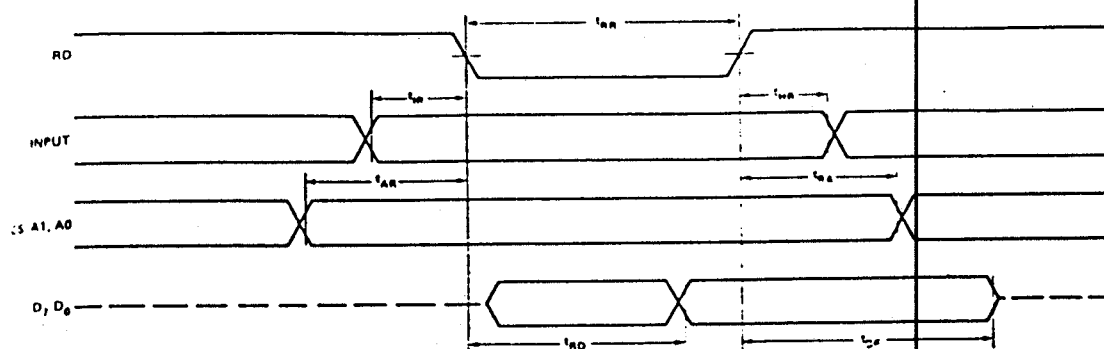
Mode 1 (Strobed Input)



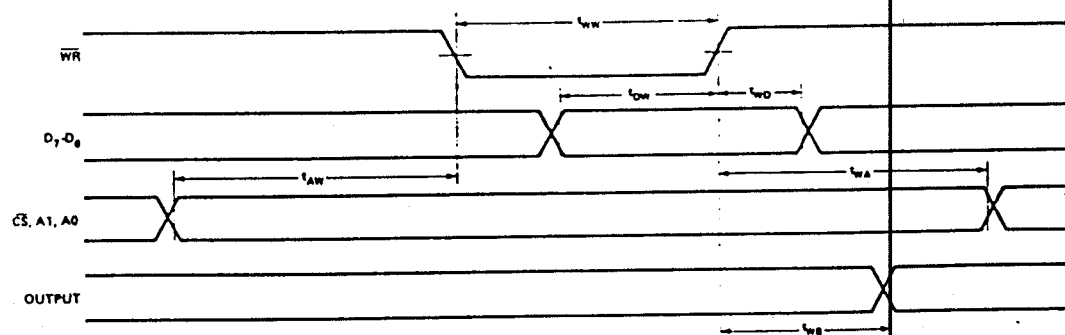
Mode 1 (Strobed Output)



### Input Waveforms For A.C. Tests



### Mode 0 (Basic Input)



### Mode 0 (Basic Output)

A.C. CHARACTERISTICS  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ;  $V_{CC} = +5V \pm 5\%$ ;  $GND = 0V$ 

## BUS PARAMETERS:

## READ:

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
$t_{AR}$	Address Stable Before READ	0		0		ns
$t_{RA}$	Address Stable After READ	0		0		ns
$t_{RR}$	READ Pulse Width	300		300		ns
$t_{RD}$	Data Valid From READ <sup>(1)</sup>		250		200	ns
$t_{DF}$	Data Float After READ	10	150	10	100	ns
$t_{RV}$	Time Between READs and/or WRITEs	850		850		ns

## WRITE:

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
$t_{AW}$	Address Stable Before WRITE	0		0		ns
$t_{WA}$	Address Stable After WRITE	20		20		ns
$t_{WW}$	WRITE Pulse Width	400		300		ns
$t_{DW}$	Data Valid to WRITE (T.E.)	100		50		ns
$t_{WD}$	Data Valid After WRITE	30		30		ns

## OTHER TIMINGS:

SYMBOL	PARAMETER	8255A		8255A-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
$t_{WB}$	WR = 1 to Output <sup>(1)</sup>		350		350	ns
$t_{IR}$	Peripheral Data Before RD	0		0		ns
$t_{HR}$	Peripheral Data After RD	0		0		ns
$t_{AK}$	ACK Pulse Width	300		300		ns
$t_{ST}$	STB Pulse Width	500		500		ns
$t_{PS}$	Per. Data Before T.E. of STB	0		0		ns
$t_{PH}$	Per. Data After T.E. of STB	180		180		ns
$t_{AD}$	ACK = 0 to Output <sup>(1)</sup>		300		300	ns
$t_{KD}$	ACK = 1 to Output Float	20	250	20	250	ns
$t_{WOB}$	WR = 1 to OBF = 0 <sup>(1)</sup>		650		550	ns
$t_{AOB}$	ACK = 0 to OBF = 1 <sup>(1)</sup>		350		350	ns
$t_{SIB}$	STB = 0 to IBF = 1 <sup>(1)</sup>		300		300	ns
$t_{RIB}$	RD = 1 to IBF = 0 <sup>(1)</sup>		300		300	ns
$t_{RIT}$	RD = 0 to INTR = 0 <sup>(1)</sup>		400		100	ns
$t_{SIT}$	STB = 1 to INTR = 1 <sup>(1)</sup>		300		300	ns
$t_{AIT}$	ACK = 1 to INTR = 1 <sup>(1)</sup>		350		350	ns
$t_{WIT}$	WR = 0 to INTR = 0 <sup>(1)</sup>		850		850	ns

Notes: 1. Test Conditions: 8255A:  $C_L = 100\text{pF}$ ; 8255A-5:  $C_L = 150\text{pF}$ .  
 2. Period of Reset pulse must be at least 50 $\mu\text{s}$  during or after power on.  
 Subsequent Reset pulse can be 500 ns min.

NOTE:  
 The 8255A-5 specifications are not final. Some parametric limits are subject to change.

# ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias. . . . . 0°C to 70°C  
 Storage Temperature . . . . . -65°C to +150°C  
 Voltage on Any Pin.  
   With Respect to Ground . . . . . -0.5V to +7V  
 Power Dissipation . . . . . 1 Watt

\*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## D.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to $70^\circ\text{C}$ , $V_{CC} = +5V \pm 5\%$ ; GND = 0V

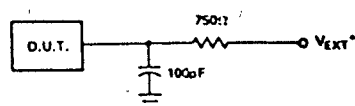
SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
$V_{IL}$	Input Low Voltage	-0.5	0.8	V	
$V_{IH}$	Input High Voltage	2.0	$V_{CC}$	V	
$V_{OL}(DB)$	Output Low Voltage (Data Bus)		0.45	V	$I_{OL} = 2.5\text{mA}$
$V_{OL}(PER)$	Output Low Voltage (Peripheral Port)		0.45	V	$I_{OL} = 1.7\text{mA}$
$V_{OH}(DB)$	Output High Voltage (Data Bus)	2.4		V	$I_{OH} = -400\mu\text{A}$
$V_{OH}(PER)$	Output High Voltage (Peripheral Port)	2.4		V	$I_{OH} = -200\mu\text{A}$
$I_{DAR}^{(1)}$	Darlington Drive Current	-1.0	-4.0	mA	$R_{EXT} = 750\Omega$ ; $V_{EXT} = 1.5V$
$I_{CC}$	Power Supply Current		120	mA	
$I_{IL}$	Input Load Current		$\pm 10$	$\mu\text{A}$	$V_{IN} = V_{CC}$ to 0V
$I_{OFL}$	Output Float Leakage		$\pm 10$	$\mu\text{A}$	$V_{OUT} = V_{CC}$ to 0V

Note 1: Available on any 8 pins from Port B and C.

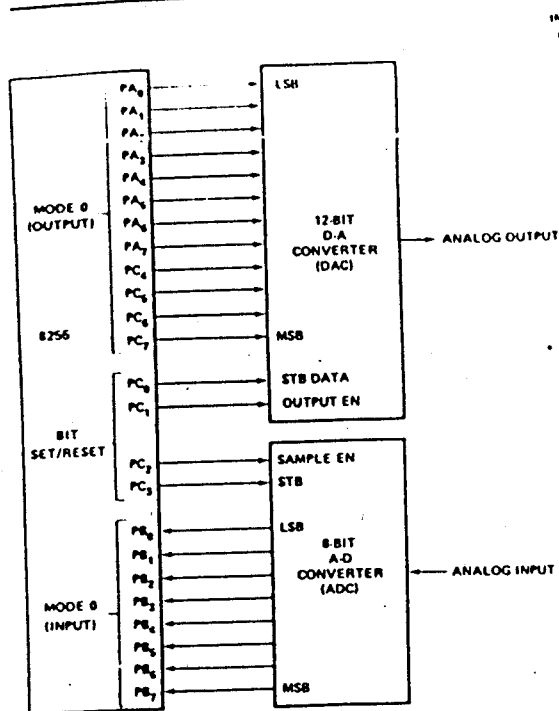
## CAPACITANCE $T_A = 25^\circ\text{C}$ ; $V_{CC} = \text{GND} = 0V$

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
$C_{IN}$	Input Capacitance			10	pF	$f_c = 1\text{MHz}$
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND

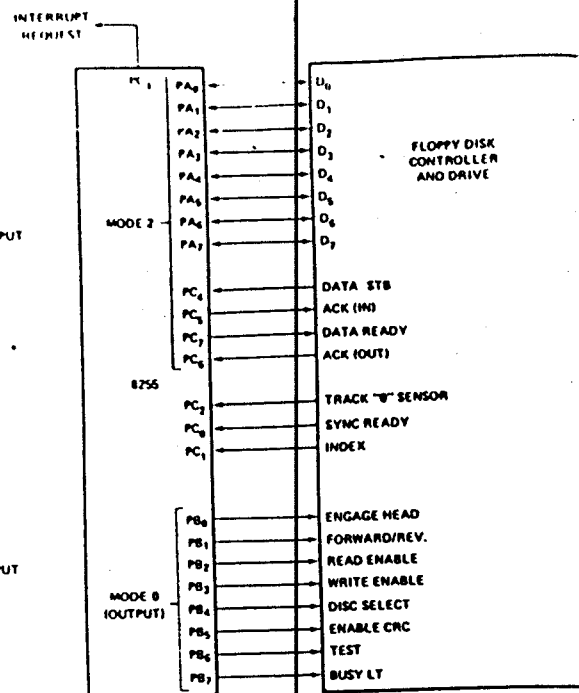
## TEST LOAD CIRCUIT (FOR DB)



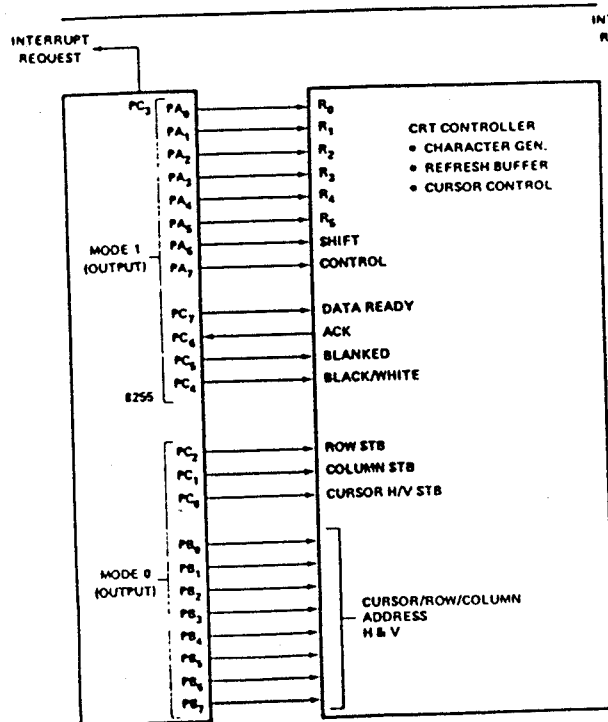
\*  $V_{EXT}$  IS SET AT VARIOUS VOLTAGES DURING TESTING TO GUARANTEE THE SPECIFICATION.



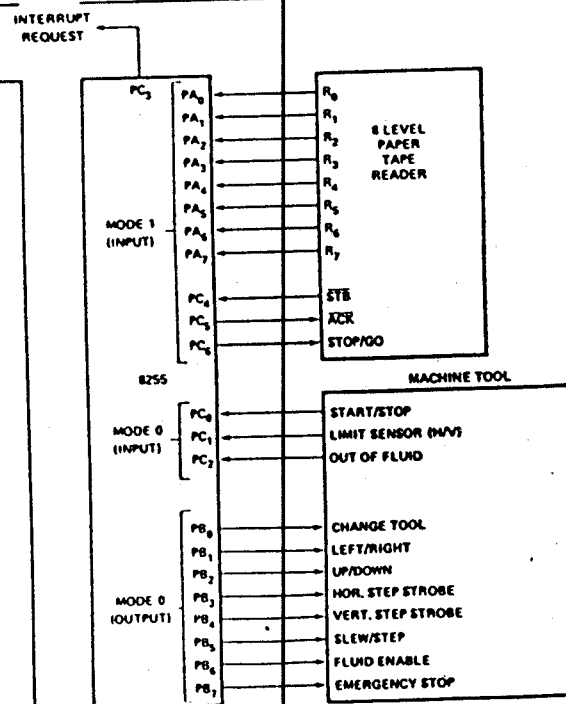
Digital to Analog, Analog to Digital



Basic Floppy Disc Interface



Basic CRT Controller Interface



Machine Tool Controller Interface



MODE DEFINITION SUMMARY TABLE

	MODE 0		MODE 1		MODE 2 GROUP A ONLY	
	IN	OUT	IN	OUT		
PA <sub>0</sub>	IN	OUT	IN	OUT	←→	
PA <sub>1</sub>	IN	OUT	IN	OUT	←→	
PA <sub>2</sub>	IN	OUT	IN	OUT	←→	
PA <sub>3</sub>	IN	OUT	IN	OUT	←→	
PA <sub>4</sub>	IN	OUT	IN	OUT	←→	
PA <sub>5</sub>	IN	OUT	IN	OUT	←→	
PA <sub>6</sub>	IN	OUT	IN	OUT	←→	
PA <sub>7</sub>	IN	OUT	IN	OUT	←→	
PB <sub>0</sub>	IN	OUT	IN	OUT	—	
PB <sub>1</sub>	IN	OUT	IN	OUT	—	
PB <sub>2</sub>	IN	OUT	IN	OUT	—	
PB <sub>3</sub>	IN	OUT	IN	OUT	—	
PB <sub>4</sub>	IN	OUT	IN	OUT	—	
PB <sub>5</sub>	IN	OUT	IN	OUT	—	
PB <sub>6</sub>	IN	OUT	IN	OUT	—	
PB <sub>7</sub>	IN	OUT	IN	OUT	—	
PC <sub>0</sub>	IN	OUT	INTR <sub>B</sub>	INTR <sub>B</sub>	I/O	
PC <sub>1</sub>	IN	OUT	IBF <sub>B</sub>	OBFB	I/O	
PC <sub>2</sub>	IN	OUT	STB <sub>B</sub>	ACK <sub>B</sub>	I/O	
PC <sub>3</sub>	IN	OUT	INTR <sub>A</sub>	INTR <sub>A</sub>	INTR <sub>A</sub>	
PC <sub>4</sub>	IN	OUT	STB <sub>A</sub>	I/O	STB <sub>A</sub>	
PC <sub>5</sub>	IN	OUT	IBFA	I/O	IBFA	
PC <sub>6</sub>	IN	OUT	I/O	ACK <sub>A</sub>	ACK <sub>A</sub>	
PC <sub>7</sub>	IN	OUT	I/O	OBFA	OBFA	

MODE 0  
OR MODE 1  
ONLY

## Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs —

All input lines can be accessed during a normal Port C read.

If Programmed as Outputs —

Bits in C upper (PC<sub>7</sub>-PC<sub>4</sub>) must be individually accessed using the bit set/reset function.

Bits in C lower (PC<sub>3</sub>-PC<sub>0</sub>) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

## Source Current Capability on Port B and Port C

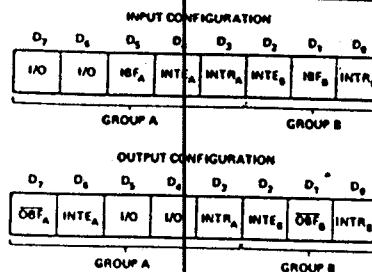
Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

## Reading Port C Status

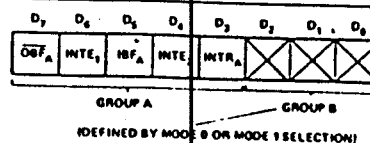
In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C

allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.



## Mode 1 Status Word Format

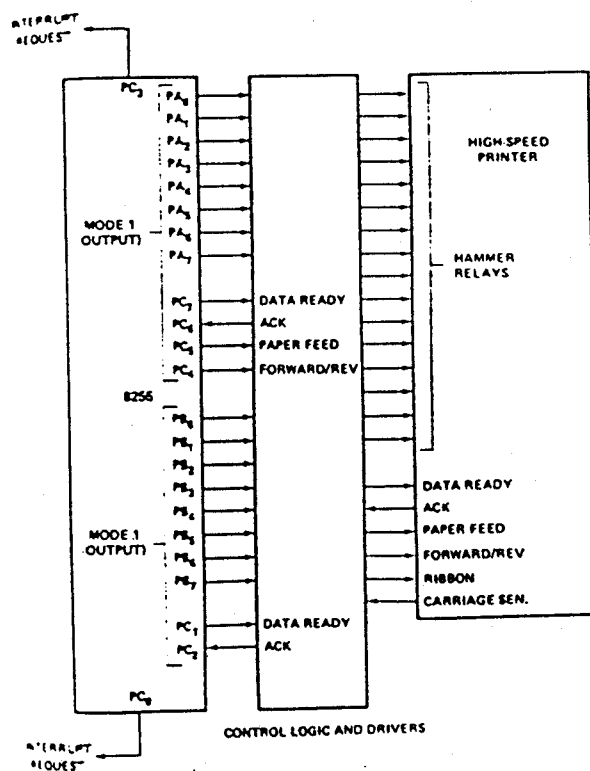


## Mode 2 Status Word Format

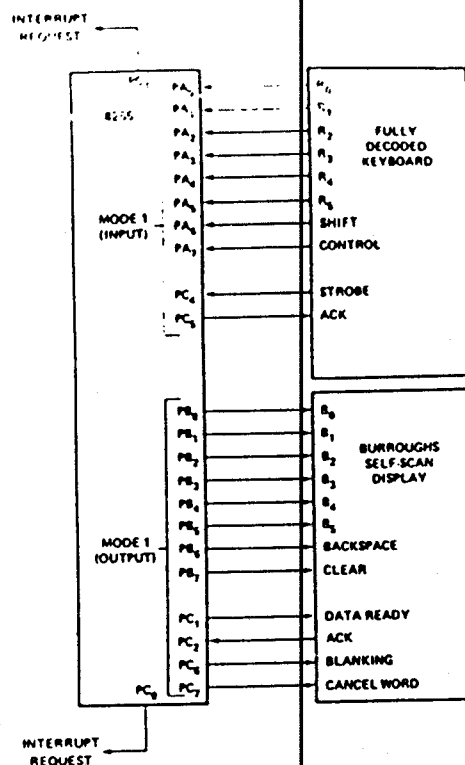
## APPLICATIONS OF THE 8255

The 8255 is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

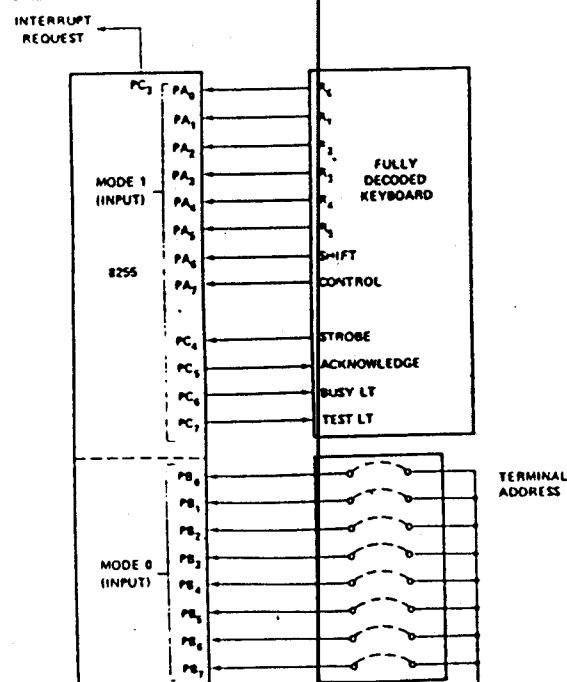
Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255 is programmed by the I/O service routine and becomes an extension of the systems software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the Detailed Operational Description, a control word can easily be developed to initialize the 8255 to exactly "fit" the application. Here are a few examples of typical applications of the 8255.



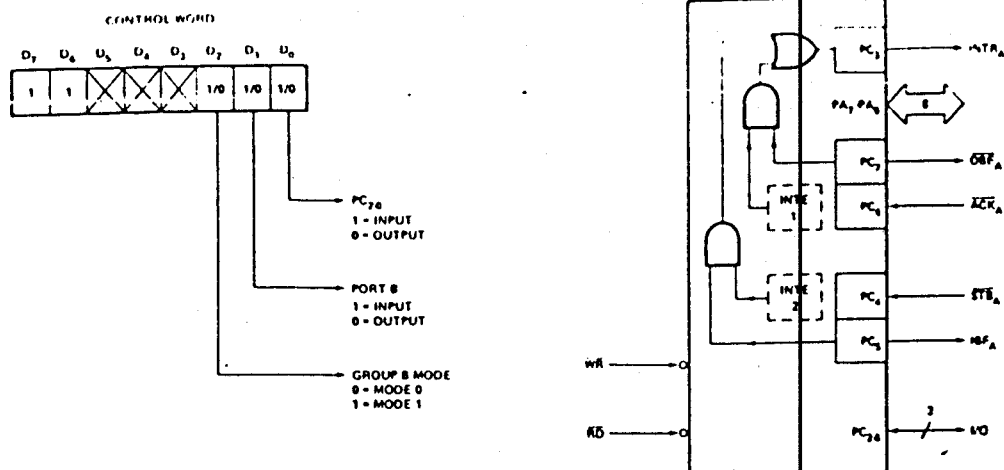
Printer Interface



Keyboard and Display Interface

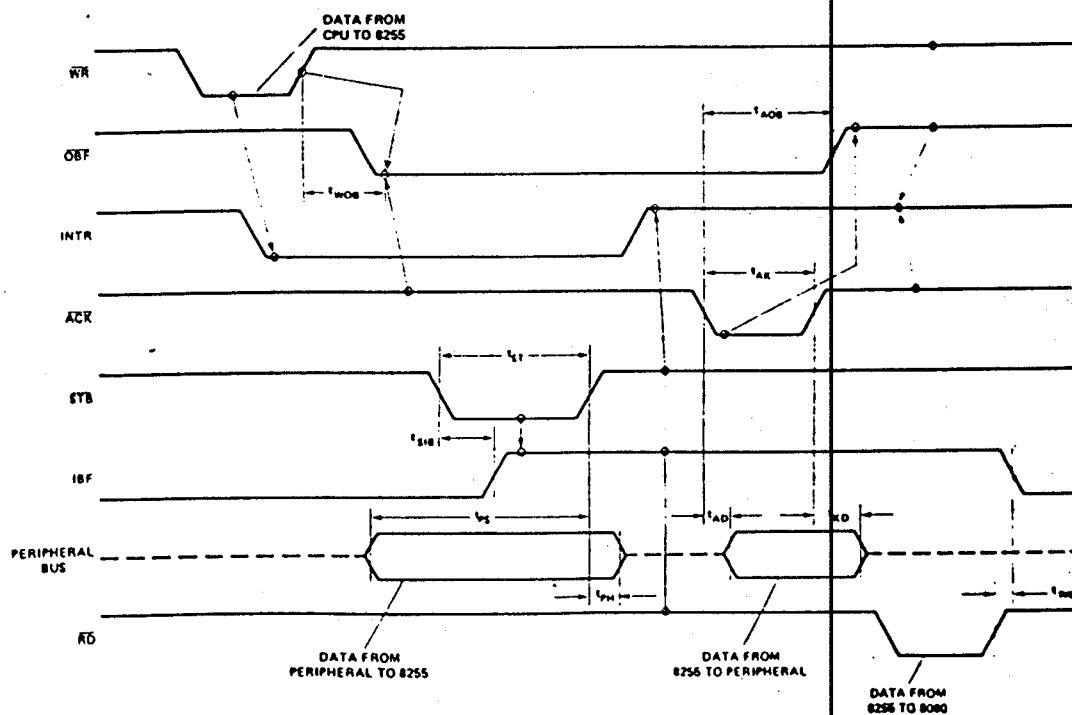


Keyboard and Terminal Address Interface



Mode 2 Control Word

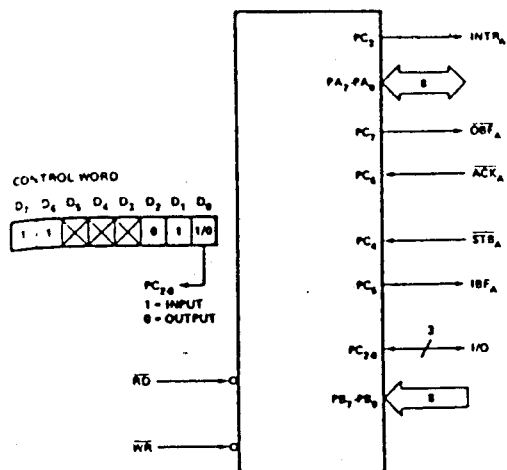
Mode 2



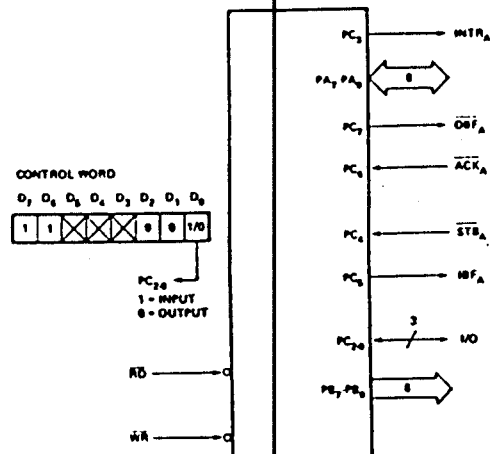
Mode 2 (Bi-directional)

NOTE: Any sequence where  $\overline{WR}$  occurs before  $\overline{ACK}$  and  $\overline{STB}$  occurs before  $\overline{RD}$  is permissible.  
 (INTR = IBF · MASK · STB · RD + OBF · MASK · ACK · WR)

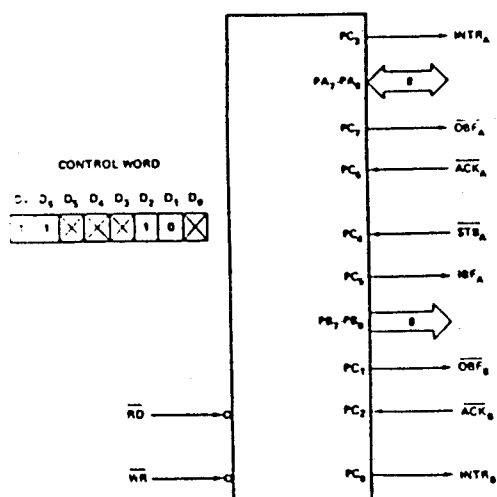
MODE 2 AND MODE 0 (INPUT)



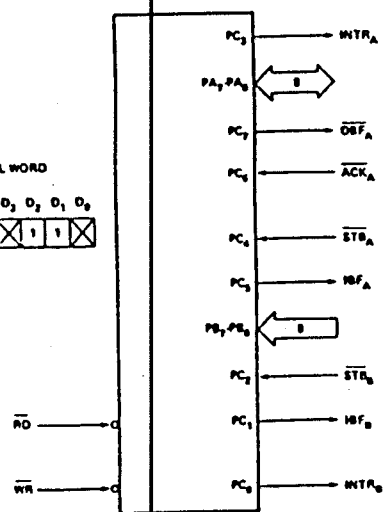
MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)



## Output Control Signal Definition

### $\overline{\text{OBF}}$ (Output Buffer Full F/F)

The  $\overline{\text{OBF}}$  output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by  $\overline{\text{ACK}}$  input being low.

### $\overline{\text{ACK}}$ (Acknowledge Input)

A "low" on this input informs the 8255 that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

### INTR (Interrupt Request)

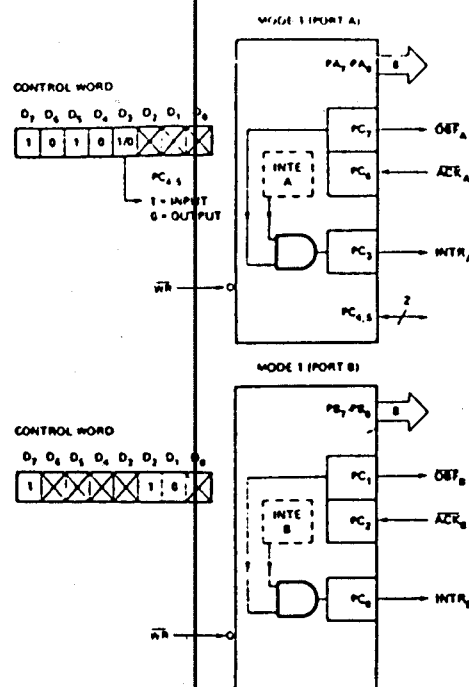
A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set by  $\overline{\text{ACK}}$  is a "one",  $\overline{\text{OBF}}$  is a "one" and INTE is a "one". It is reset by the falling edge of WR.

#### INTE A

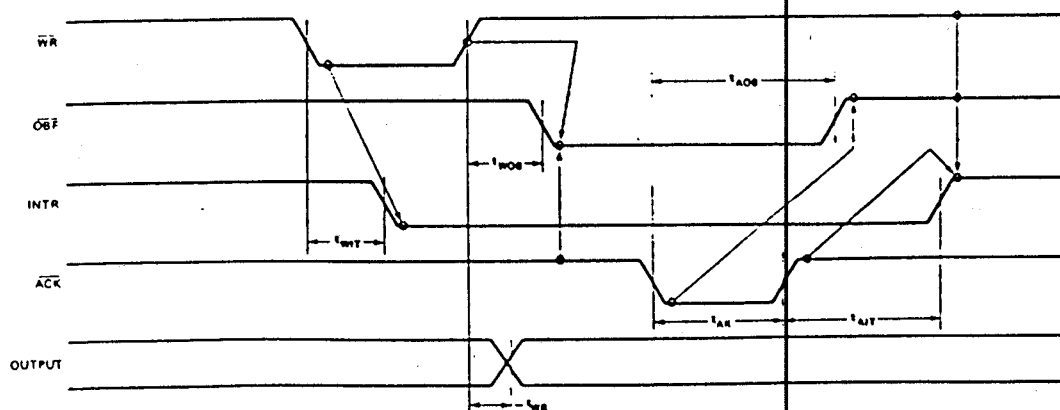
Controlled by bit set/reset of PC<sub>6</sub>.

#### INTE B

Controlled by bit set/reset of PC<sub>2</sub>.



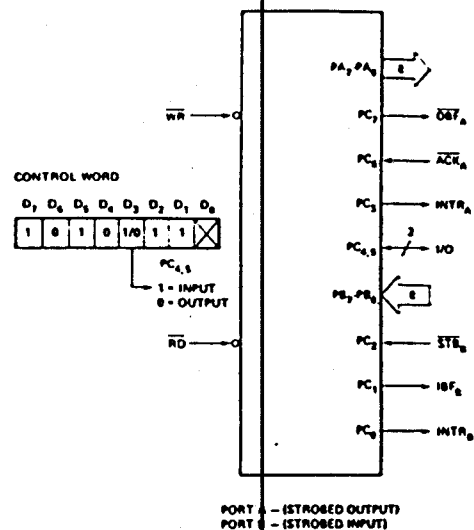
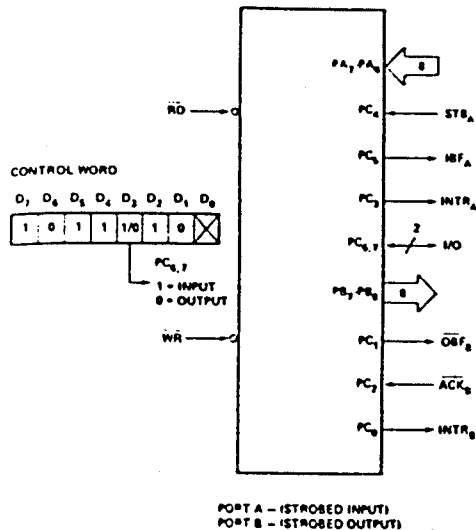
Mode 1 Output



Mode 1 (Strobed Output)

### Combinations of Mode 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.



### Operating Modes

#### Mode 2 (Strobed Bi-Directional Bus I/O)

This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to Mode 1. Interrupt generation and enable/disable functions are also available.

##### Mode 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

#### Bi-Directional Bus I/O Control Signal Definition

##### INTR (Interrupt Request)

A high on this output can be used to interrupt the CPU for both input or output operations.

#### Output Operations

##### OBF (Output Buffer Full)

The OBF output will go "low" to indicate that the CPU has written data out to Port A.

##### ACK (Acknowledge)

A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high-impedance state.

##### INTE 1 (The INTE Flip-Flop associated with OBF)

Controlled by bit set/reset of PC<sub>6</sub>.

#### Input Operations

##### STB (Strobe Input)

A "low" on this input loads data into the input latch.

##### IBF (Input Buffer Full F/F)

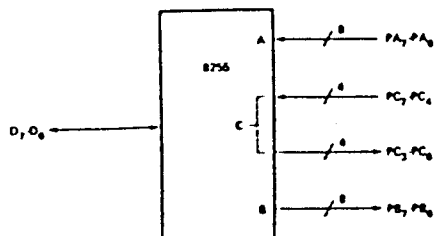
A "high" on this output indicates that data has been loaded into the input latch.

##### INTE 2 (The INTE Flip-Flop associated with IBF)

Controlled by bit set/reset of PC<sub>4</sub>.

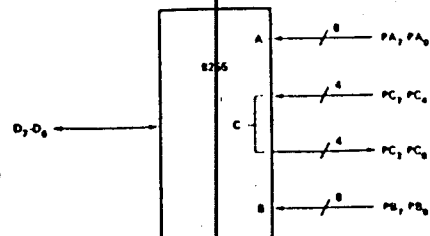
CONTROL WORD #12

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	1	0	0	0



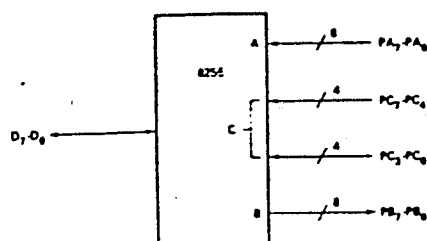
CONTROL WORD #14

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	1	0	1	0



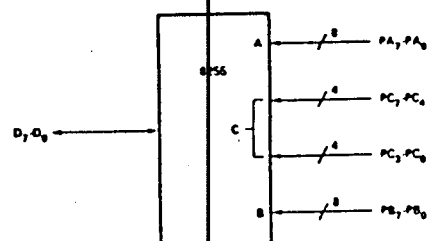
CONTROL WORD #13

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	1	0	0	1



CONTROL WORD #15

D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	1	1	0	1	1



## Operating Modes

### Mode 1 (Strobed Input/Output)

This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

### Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

## ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

### PRINCIPLES OF OPERATION

The ADC0808 and ADC0809 each consists of an analog signal multiplexer, an 8-bit successive-approximation converter, and related control and output circuitry.

#### multiplexer

The analog multiplexer selects 1 of 8 single-ended input channels as determined by the address decoder. Address load control loads the address code into the decoder on a low-to-high transition. The output latch is reset by the positive-going edge of the start pulse. Sampling also starts with the positive-going edge of the start pulse and lasts for 32 clock periods. The conversion process may be interrupted by a new start pulse before the end of 64 clock periods. The previous data will be lost if a new start of conversion occurs before the 64th clock pulse. Continuous conversion may be accomplished by connecting the End-of-Conversion output to the start input. If used in this mode an external pulse should be applied after power up to assure start up.

#### converter

The CMOS threshold detector in the successive-approximation conversion system determines each bit by examining the charge on a series of binary-weighted capacitors (Figure 1). In the first phase of the conversion process, the analog input is sampled by closing switch  $S_C$  and all  $S_T$  switches, and by simultaneously charging all the capacitors to the input voltage.

In the next phase of the conversion process, all  $S_T$  and  $S_C$  switches are opened and the threshold detector begins identifying bits by identifying the charge (voltage) on each capacitor relative to the reference voltage. In the switching sequence, all eight capacitors are examined separately until all 8 bits are identified, and then the charge-convert sequence is repeated. In the first step of the conversion phase, the threshold detector looks at the first capacitor (weight = 128). Node 128 of this capacitor is switched to the reference voltage, and the equivalent nodes of all the other capacitors on the ladder are switched to REF-. If the voltage at the summing node is greater than the trip-point of the threshold detector (approximately one-half the  $V_{CC}$  voltage), a bit is placed in the output register, and the 128-weight capacitor is switched to REF-. If the voltage at the summing node is less than the trip point of the threshold detector, this 128-weight capacitor remains connected to REF+ through the remainder of the capacitor-sampling (bit-counting) process. The process is repeated for the 64-weight capacitor, the 32-weight capacitor, and so forth down the line, until all bits are counted.

With each step of the capacitor-sampling process, the initial charge is redistributed among the capacitors. The conversion process is successive approximation, but relies on charge redistribution rather than a successive-approximation register (and reference DAC) to count and weigh the bits from MSB to LSB.

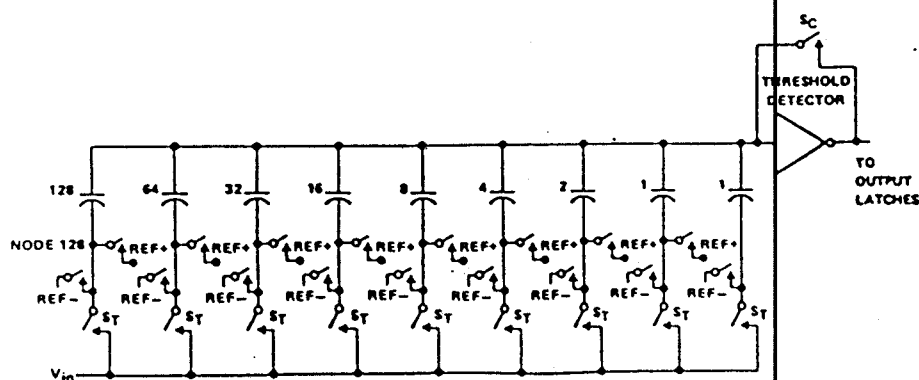


FIGURE 1. SIMPLIFIED MODEL OF THE SUCCESSIVE-APPROXIMATION SYSTEM

2

Data Acquisition Circuits



**ADC0808, ADC0809**  
**CMOS ANALOG-TO-DIGITAL CONVERTERS**  
**WITH 8-CHANNEL MULTIPLEXERS**

operating characteristics,  $T_A = 25^\circ\text{C}$ ,  $V_{CC} = V_{REF+} = 5\text{ V}$ ,  $V_{REF-} = 0\text{ V}$ ,  $f_{\text{clock}} = 640\text{ kHz}$   
(unless otherwise noted)

**2**

Data Acquisition Circuits

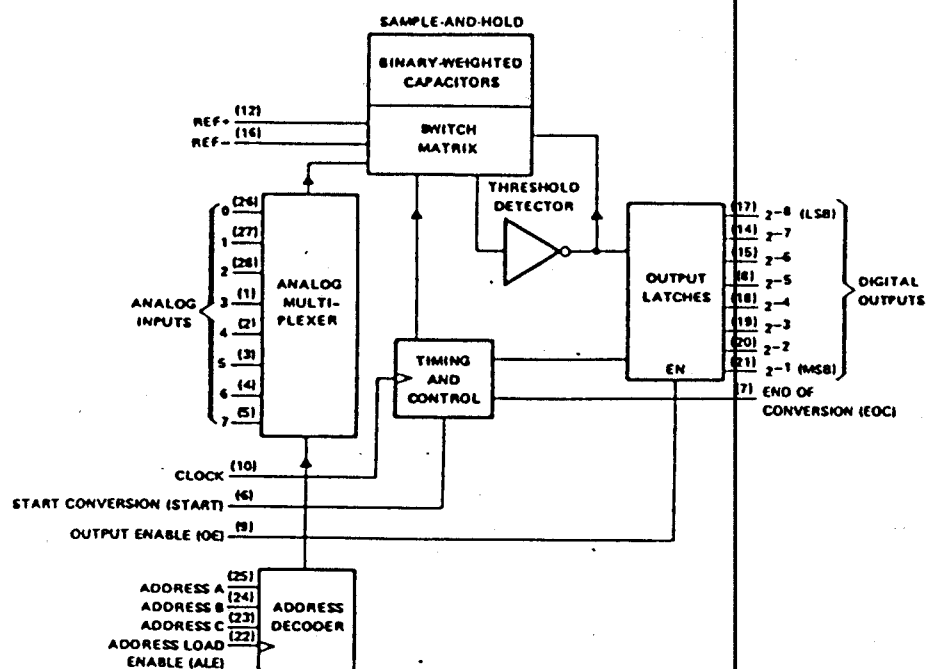
PARAMETER		TEST CONDITIONS	ADC0808			ADC0809			UNIT
			MIN	TYP <sup>1</sup>	MAX	MIN	TYP <sup>1</sup>	MAX	
$k_{SVS}$	Supply voltage sensitivity	$V_{CC} = V_{ref+} = 4.75\text{ V to } 5.25\text{ V}$ , $T_A = 40^{\circ}\text{C to } 85^{\circ}\text{C}$ . See Note 4	$\pm 0.05$			$\pm 0.05$			%/V
	Linearity error (see Note 5)		$\pm 0.25$			$\pm 0.5$			LSB
	Zero error (see Note 6)		$\pm 0.25$			$\pm 0.25$			LSB
	Total unadjusted error (See Note 7)	$T_A = 25^{\circ}\text{C}$	$\pm 0.25$			$\pm 0.5$			LSB
		$T_A = -40^{\circ}\text{C to } 85^{\circ}\text{C}$	$\pm 0.75$			$\pm 1.25$			
		$T_A = 0^{\circ}\text{C to } 70^{\circ}\text{C}$				$\pm 1$			
$t_{en}$	Output enable time	$C_L = 50\text{ pF}$ , $R_L = 10\text{ k}\Omega$	80	250		80	250		ns
$t_{dis}$	Output disable time	$C_L = 10\text{ pF}$ , $R_L = 10\text{ k}\Omega$	105	250		105	250		ns
$t_{conv}$	Conversion time	See Note 8	90	100	116	90	100	116	$\mu\text{s}$
$t_{d(EOC)}$	Delay time, end of conversion output	See Notes 8 and 9	0 14.5			0 14.5			$\mu\text{s}$

<sup>1</sup>Typical values for all except supply voltage sensitivity are at  $V_{CC} = 5\text{ V}$ , and all are at  $T_A = 25^\circ\text{C}$ .

- NOTES: 4. Supply voltage sensitivity relates to the ability of an analog-to-digital converter to maintain accuracy as the supply voltage varies. The supply and  $V_{\text{ref}+}$  are varied together and the change in accuracy is measured with respect to full-scale.
5. Linearity error is the maximum deviation from a straight line through the end points of the A/D transfer characteristic.
6. Zero error is the difference between 00000000 and the converted output for zero input voltage; full-scale error is the difference between 11111111 and the converted output for full-scale input voltage.
7. Total unadjusted error is the maximum sum of linearity error, zero error, and full-scale error.
8. Refer to the operating sequence diagram.
9. For clock frequencies other than 640 kHz,  $t_{\text{d(EOC)}}$  maximum is 8 clock periods plus 2  $\mu\text{s}$ .

2

## Data Acquisition Circuits



### MULTIPLEXER FUNCTION TABLE

MULTIFLEXER FUNCTION TABLE				
INPUTS				SELECTED ANALOG CHANNEL
ADDRESS			ADDRESS STROBE	
C	B	A		
L	L	L	↑	0
L	L	H	↑	1
L	H	L	↑	2
L	H	H	↑	3
H	L	L	↑	4
H	L	H	↑	5
H	H	L	↑	6
H	H	H	↑	7

H = high level, L = low level  
↑ = low-to-high transition

**ADC0808, ADC0809**  
**CMOS ANALOG-TO-DIGITAL CONVERTERS**  
**WITH 8-CHANNEL MULTIPLEXERS**

electrical characteristics over recommended operating free-air temperature range,  $V_{CC} = 4.75 \text{ V}$  to  $5.25 \text{ V}$  (unless otherwise noted)

total device

PARAMETER		TEST CONDITIONS	MIN	TYP <sup>†</sup>	MAX	UNIT
$V_{OH}$	High-level output voltage	$I_O = -380 \mu\text{A}$	$V_{CC} - 0.4$			V
$V_{OL}$	Low-level output voltage	Data outputs $I_O = 1.8 \text{ mA}$			0.45	V
		End of conversion $I_O = 1.2 \text{ mA}$			0.45	
$I_{OZ}$	Off-state (high-impedance-state) output current	$V_O = V_{CC}$			3	$\mu\text{A}$
		$V_O = 0$			-3	
$I_i$	Control input current at maximum input voltage	$V_i = 15 \text{ V}$			1	$\mu\text{A}$
$I_{iL}$	Low-level control input current	$V_i = 0$			-1	$\mu\text{A}$
$I_{CC}$	Supply current	$f_{\text{clock}} = 640 \text{ kHz}$		0.3	3	mA
$C_i$	Input capacitance, control inputs	$T_A = 25^\circ\text{C}$		10	15	pF
$C_o$	Output capacitance, data outputs	$T_A = 25^\circ\text{C}$		10	15	pF
Resistance from pin 12 to pin 16				1000		k $\Omega$

analog multiplexer

PARAMETER		TEST CONDITIONS	MIN	TYP <sup>†</sup>	MAX	UNIT
$I_{on}$	Channel on-state current (see Note 3)	$V_i = 5 \text{ V}, f_{\text{clock}} = 640 \text{ kHz}$			2	$\mu\text{A}$
		$V_i = 0, f_{\text{clock}} = 640 \text{ kHz}$			-2	
$I_{off}$	Channel off-state current	$V_{CC} = 5 \text{ V}, T_A = 25^\circ\text{C}$			10	nA
		$V_i = 5 \text{ V}$			200	
		$V_i = 0$			-10	-200
		$V_{CC} = 5 \text{ V}$			1	
		$V_i = 5 \text{ V}$			-1	$\mu\text{A}$
		$V_i = 0$				

<sup>†</sup>Typical values are at  $V_{CC} = 5 \text{ V}$  and  $T_A = 25^\circ\text{C}$ .

NOTE 3: Channel on-state current is primarily due to the bias current into or out of the threshold detector, and it varies directly with clock frequency.

2

Data Acquisition Circuits

# ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

2

Data Acquisition Circuits

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, $V_{CC}$ (see Note 1)	6.5 V
Input voltage range: control inputs	-0.3 to 15 V
all other inputs	-0.3 V to $V_{CC} + 0.3$ V
Operating free-air temperature range	-40°C to 85°C
Storage temperature range	-65°C to 150°C
Lead temperature 1.6 mm (1/16 inch) from case for 10 seconds	260°C

NOTE 1: All voltage values are with respect to network ground terminal.

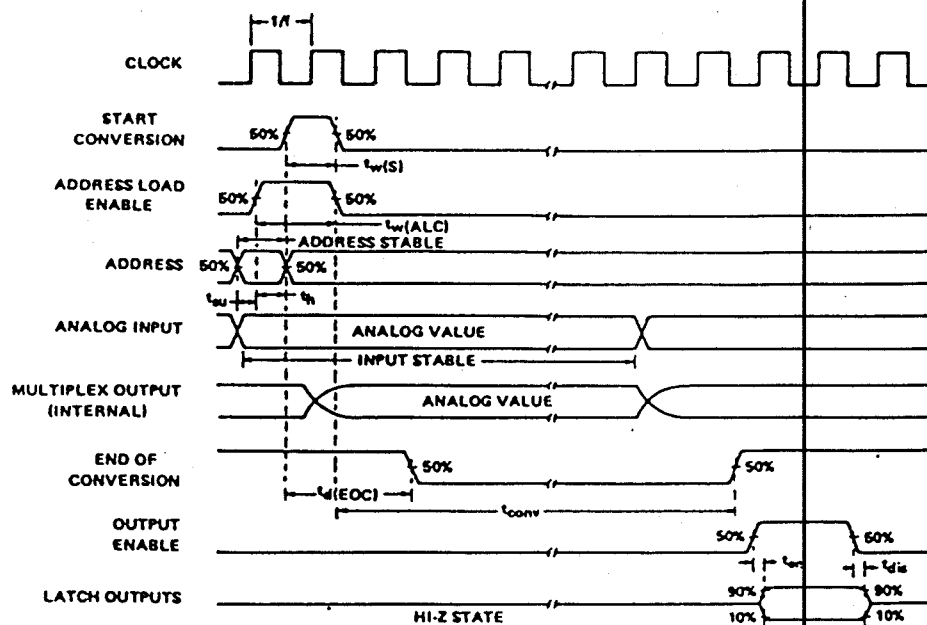
## recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, $V_{CC}$	4.5	5	6	V
Positive reference voltage, $V_{ref+}$ (see Note 2)		$V_{CC}$	$V_{CC} + 0.1$	V
Negative reference voltage, $V_{ref-}$		0	-0.1	V
Differential reference voltage, $V_{ref+} - V_{ref-}$		5		V
High-level input voltage, $V_{IH}$	$V_{CC} - 1.5$			V
Low-level input voltage, $V_{IL}$			1.5	V
Start pulse duration, $t_w(S)$	200			ns
Address load control pulse duration, $t_w(ALC)$	200			ns
Address setup time, $t_{su}$	50			ns
Address hold time, $t_h$	50			ns
Clock frequency, $f_{clock}$	10	640	1280	kHz
Operating free-air temperature, $T_A$	-40		85	°C

NOTE 2: Care must be taken that this rating is observed even during power-up.

# CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

operating sequence

2  
Data Acquisition Circuits

# ADC0808, ADC0809 CMOS ANALOG-TO-DIGITAL CONVERTERS WITH 8-CHANNEL MULTIPLEXERS

02642, JUNE 1981 REVISED FEBRUARY 1986

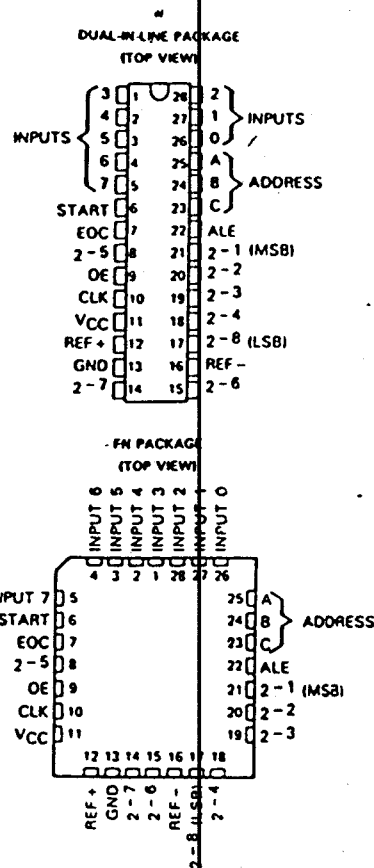
- Total Unadjusted Error . . .  $\pm 0.75$  LSB Max for ADC0808 and  $\pm 1.25$  LSB Max for ADC0809
- Resolution of 8 Bits
- 100  $\mu$ s Conversion Time
- Ratiometric Conversion
- Guaranteed Monotonicity
- No Missing Codes
- Easy Interface with Microprocessors
- Latched 3-State Outputs
- Latched Address Inputs
- Single 5-Volt Supply
- Low Power Consumption
- Designed to be Interchangeable with National Semiconductor ADC0808, ADC0809

## description

The ADC0808 and ADC0809 are monolithic CMOS devices with an 8-channel multiplexer, an 8-bit analog-to-digital (A/D) converter, and microprocessor-compatible control logic. The 8-channel multiplexer can be controlled by a microprocessor through a 3-bit address decoder with address load to select any one of eight single-ended analog switches connected directly to the comparator. The 8-bit A/D converter uses the successive-approximation conversion technique featuring a high-impedance threshold detector, a switched-capacitor array, a sample-and-hold, and a successive-approximation register (SAR). Detailed information on interfacing to most popular microprocessors is readily available from the factory.

The comparison and converting methods used eliminate the possibility of missing codes, nonmonotonicity, and the need for zero or full-scale adjustment. Also featured are latched 3-state outputs from the SAR and latched inputs to the multiplexer address decoder. The single 5-volt supply and low power requirements make the ADC0808 and ADC0809 especially useful for a wide variety of applications. Ratiometric conversion is made possible by access to the reference voltage input terminals.

The ADC0808 and ADC0809 are characterized for operation from  $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ .



2

Data Acquisition Circuits

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS  
INSTRUMENTS  
POST OFFICE BOX 655012 • DALLAS, TEXAS 75265

Copyright © 1983, Texas Instruments Incorporated

2-21

## ADC0804I, ADC0804C 8-BIT ANALOG-TO-DIGITAL CONVERTER WITH DIFFERENTIAL INPUTS

2

Data Acquisition Circuits

### PRINCIPLES OF OPERATION

The ADC0804 contains a circuit equivalent to a 256-resistor network. Analog switches are sequenced by successive approximation logic to match an analog differential input voltage ( $V_{IN+} - V_{IN-}$ ) to a corresponding tap on the 256-resistor network. The most-significant bit (MSB) is tested first. After eight comparisons (64 clock periods), an eight-bit binary code (1111 1111 = full scale) is transferred to an output latch and the interrupt ( $\overline{INTR}$ ) output goes low. The device can be operated in a free-running mode by connecting the  $\overline{INTR}$  output to the write ( $\overline{WR}$ ) input and holding the conversion start ( $\overline{CS}$ ) input at a low level. To ensure start-up under all conditions, a low-level  $\overline{WR}$  input is required during the power-up cycle. Taking  $\overline{CS}$  low anytime after that will interrupt a conversion in process.

When the  $\overline{WR}$  input goes low, the ADC0804 successive approximation register (SAR) and eight-bit shift register are reset. As long as both  $\overline{CS}$  and  $\overline{WR}$  remain low, the ADC0804 will remain in a reset state. One to eight clock periods after  $\overline{CS}$  or  $\overline{WR}$  makes a low-to-high transition, conversion starts.

When the  $\overline{CS}$  and  $\overline{WR}$  inputs are low, the start flip-flop is set and the interrupt flip-flop and eight-bit register are reset. The next clock pulse transfers a logic high to the output of the start flip-flop. The logic high is ANDed with the next clock pulse placing a logic high on the reset input of the start flip-flop. If either  $\overline{CS}$  or  $\overline{WR}$  have gone high, the set signal to the start flip-flop is removed causing it to be reset. A logic high is placed on the D input of the eight-bit shift register and the conversion process is started. If the  $\overline{CS}$  and  $\overline{WR}$  inputs are still low, the start flip-flop, the eight-bit shift register, and the SAR remain reset. This action allows for wide  $\overline{CS}$  and  $\overline{WR}$  inputs with conversion starting from one to eight clock periods after one of the inputs goes high.

When the logic high input has been clocked through the eight-bit shift register, completing the SAR search, it is applied to an AND gate controlling the output latches and to the D input of a flip-flop. On the next clock pulse, the digital word is transferred to the three-state output latches and the interrupt flip-flop is set. The output of the interrupt flip-flop is inverted to provide an  $\overline{INTR}$  output that is high during conversion and low when the conversion is completed.

When a low is at both the  $\overline{CS}$  and  $\overline{RD}$  inputs, an output is applied to the DB0 through DB7 outputs and the interrupt flip-flop is reset. When either the  $\overline{CS}$  or  $\overline{RD}$  inputs return to a high state, the DB0 through DB7 outputs are disabled (returned to the high-impedance state). The interrupt flip-flop remains reset.

$\overline{RD}$  is a tristate signal which indicates that the CPU wishes to read data from either memory or an I/O device, as identified by  $\overline{MREQ}$  or  $\overline{IORQ}$ .

$\overline{WR}$  is a tristate control signal which indicates that the CPU wishes to write data to memory or an I/O device as indicated by  $\overline{MREQ}$  and  $\overline{IORQ}$ . Some Z80 I/O devices have no  $\overline{WR}$  input. These devices assume a Write operation when  $\overline{IORQ}$  is low and  $\overline{RD}$  is high.  $\overline{RD}$  low specifies a Read operation.

The various ways in which the three control signals,  $\overline{M1}$ ,  $\overline{IORQ}$ , and  $\overline{RD}$ , may be interpreted are summarized in Table 7-5, which occurs in the description of the Z80 PIO device.

$\overline{RFSH}$  is a control signal used to refresh dynamic memories. When  $\overline{RFSH}$  is output low, the current  $\overline{MREQ}$  signal should be used to refresh dynamic memory, as addressed by the lower seven bits of the Address Bus, A0 - A6.

Next, we will describe CPU control signals.

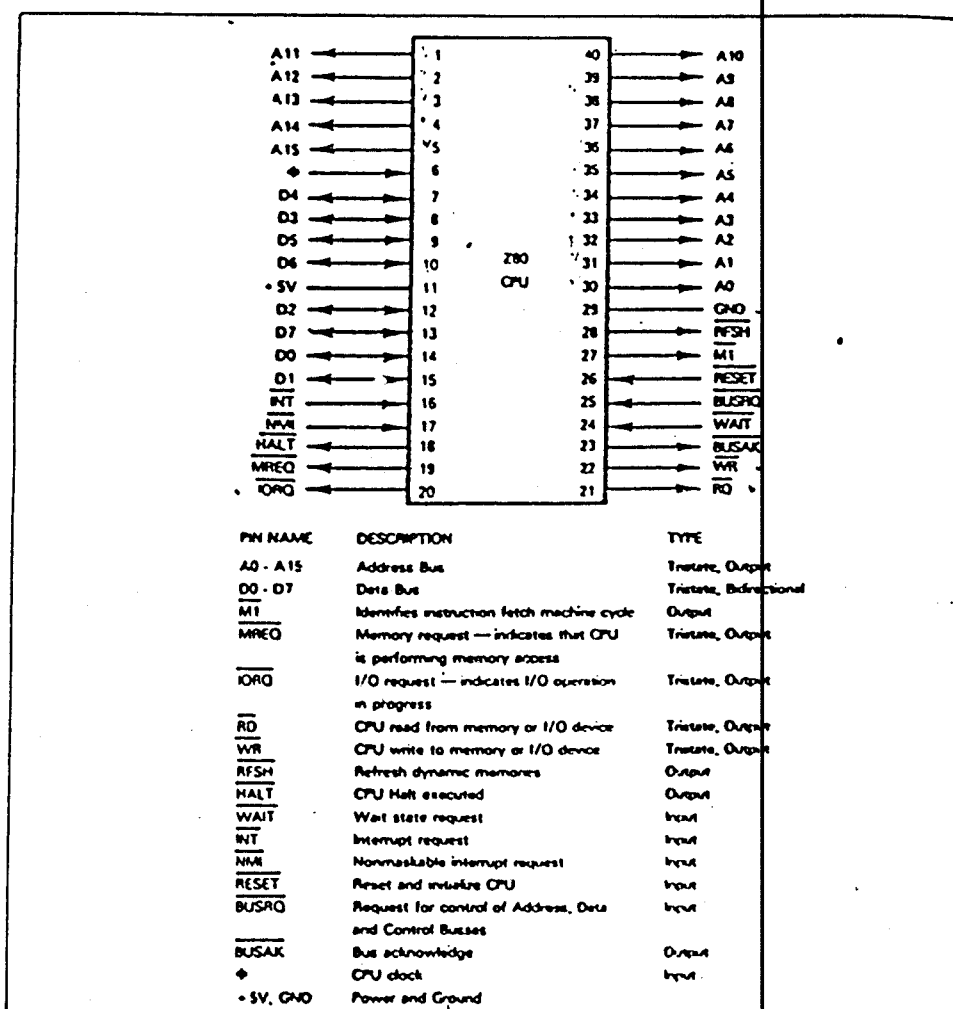


Figure 7-4. Z80 CPU Signals and Pin Assignments



**HALT** is output low following execution of a Halt instruction. The CPU now enters a Halt state during which it continuously re-executes a NOP instruction in order to maintain memory refresh activity. A Halt can only be terminated with an interrupt.

**WAIT** is equivalent to the 8080A READY input. External logic which cannot respond to a CPU access request within the allowed time interval extends the time interval by pulling the **WAIT** input low. In response to **WAIT** low, the Z80 enters a Wait state during which the CPU inserts an integral number of clock periods. Taken together, these clock periods constitute a Wait state.

**INT** and **NMI** are two interrupt request inputs. The difference between these two signals is that **NMI** has higher priority and cannot be disabled.

There are two Bus control signals.

**RESET** is a standard reset control input. When the Z80 is reset, this is what happens:

The Program Counter, IV and R registers' contents are all set to zero.

Interrupt requests via **INT** are disabled.

All tristate bus signals are floated.

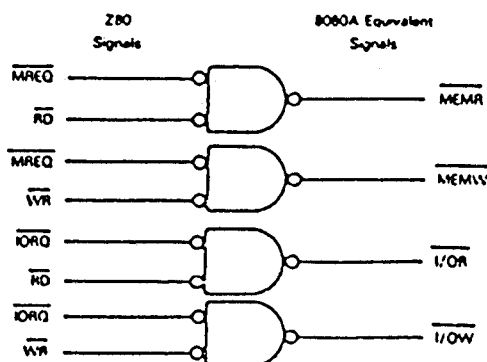
**BUSRQ** and **BUSAK** are bus request and acknowledge signals. In order to perform any kind of DMA operation, external logic must acquire control of the microcomputer System Bus. This is done by inputting **BUSRQ** low, at the conclusion of the current machine cycle, the Z80 CPU will float all tristate bus lines and will acknowledge the bus request by outputting **BUSAK** low.

## Z80 - 8080A SIGNAL COMPATIBILITY

If you are designing a new product around the Z80 CPU, then questions of Z80 - 8080A signal compatibility are irrelevant; you will design for the CPU on hand.

If you are replacing an 8080A with a Z80, then it would be helpful to have some type of lookup table which directly relates 8080A signals to Z80 signals. Unfortunately, such a lookup table cannot easily be created. The problem is that the Z80 is an implementation of three devices: the 8080A CPU, the 8224 Clock, and 8228 System Controller, but there are very many 8080A configurations that do not include an 8228 System Controller.

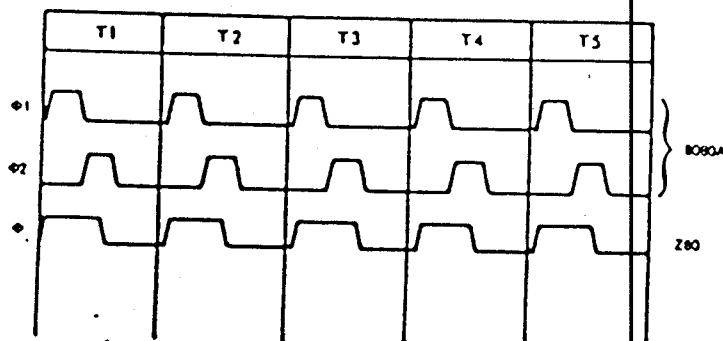
Probably the most important conceptual difference between the Z80 and 8080A involves read and write control signals. The 8228 System Controller develops four discrete control signals for memory read, memory write, I/O read and I/O write. The Z80 has a general read and a general write, coupled with an I/O select and a memory select. By adding logic, it would be easy enough to generate the four discrete 8080A signals from the two Z80 signal pairs. Here is one elementary possibility:



**Z80 CPU  
CONTROL  
SIGNALS**

**Z80 BUS  
CONTROL  
SIGNALS**

Z80 clock signals are also far simpler than the 8080A equivalent. Where the 8080A uses two clock signals the Z80 uses one. Clock logic may be compared as follows.



### INSTRUCTION FETCH EXECUTION SEQUENCES

As compared to the 8080A, Z80 instruction timing is marvelously simple. Gone is the SYNC pulse and the decoding of Data Bus for status. Every instruction's timing degenerates into an instruction fetch, optionally followed by memory or I/O read or write. Add to this a few variations for Wait state, interrupt acknowledge and bus floating and you are done.

Let us begin by looking at an instruction fetch. Timing is illustrated in Figure 7-5. Look at the instruction fetch timing in the 8080A chapter to obtain an immediate comparison of the Z80 and the 8080A.

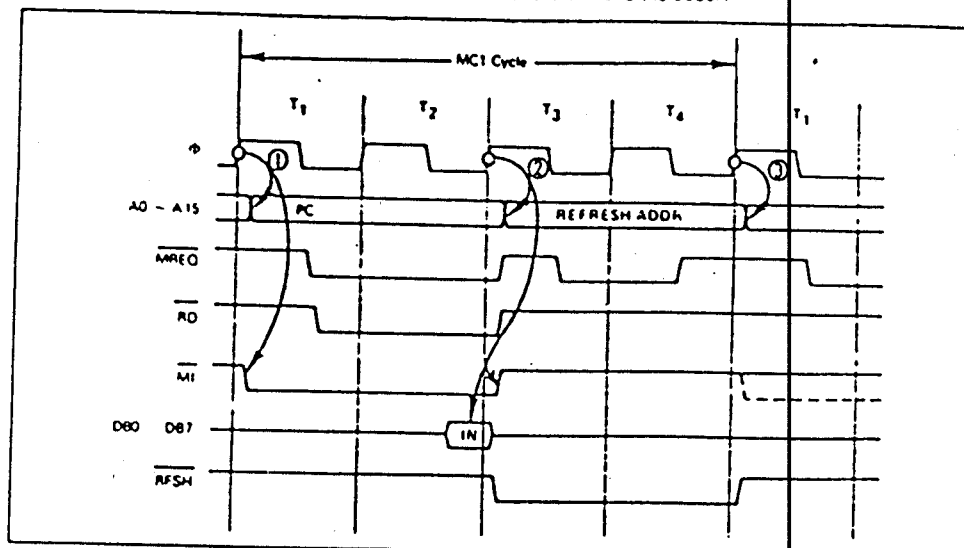


Figure 7-5 Z80 Instruction Fetch Sequence

Referring to Figure 7-5, note that the instruction fetch cycle is identified by  $\overline{M1}$  output low during  $T_1$  and  $T_2$  (①). Since there is no status on the Data Bus to worry about, the Program Counter contents are output immediately on the Address Bus and stay stable for the duration of  $T_1$  and  $T_2$ .

Since an instruction fetch is also a memory operation,  $\overline{MRQ}$  and  $\overline{MW}$  controls are both output low. This occurs half-way through  $T_1$ , at which time the Address Bus will stabilize. The falling edges of  $\overline{MRQ}$  and  $\overline{MW}$  can therefore be used to select a memory device and strobe data out. The CPU polls data on the Data Bus at the rising edge of the  $T_3$  clock (②).

Clock periods  $T_3$  and  $T_4$  of the instruction fetch machine cycle are used by the Z80 CPU for internal operations. These clock periods are also used to refresh dynamic memory. As soon as the Program Counter contents are taken off the Address Bus (②), the refresh address from the Refresh register is output on lines A0 - A6 of the Address Bus. This address stays on the Address Bus until the conclusion of  $T_4$  (③).

Since a memory refresh is a memory access operation,  $\overline{MREQ}$  is again output low. However, it is accompanied by  $\overline{RFSH}$  rather than  $\overline{RD}$  low. Thus memory reference logic does not attempt to read data during a refresh cycle.

### A MEMORY READ OPERATION

Memory interface logic responds to an instruction fetch and a memory read in exactly the same way. There are, however, a few differences between memory read and instruction fetch timing. Memory read timing is illustrated in Figure 7-6. The principal difference to note is that during a memory read operation, the data is sampled on the falling edge of the  $T_3$  clock pulse, whereas during an instruction fetch it is sampled on the rising edge of this clock pulse. Also a normal memory read machine cycle will consist of three clock periods, while the normal instruction fetch consists of four clock periods. Remember also that the Z80 identifies an instruction fetch machine cycle by outputting  $\overline{MT}$  low during the first two clock periods of the instruction fetch machine cycle.

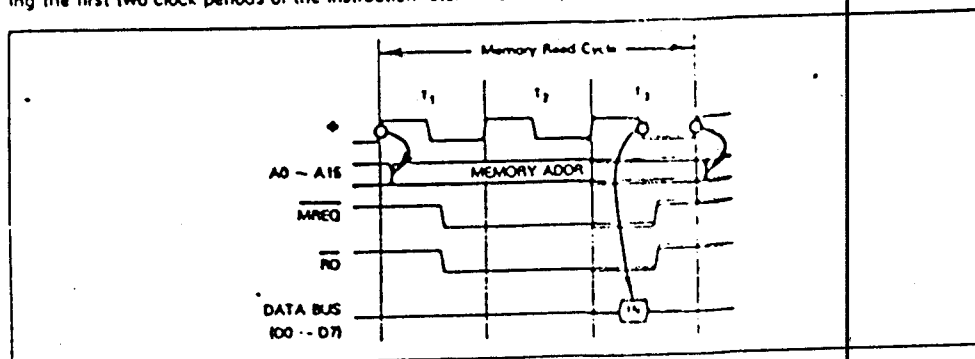


Figure 7-6 Z80 Memory Read Timing

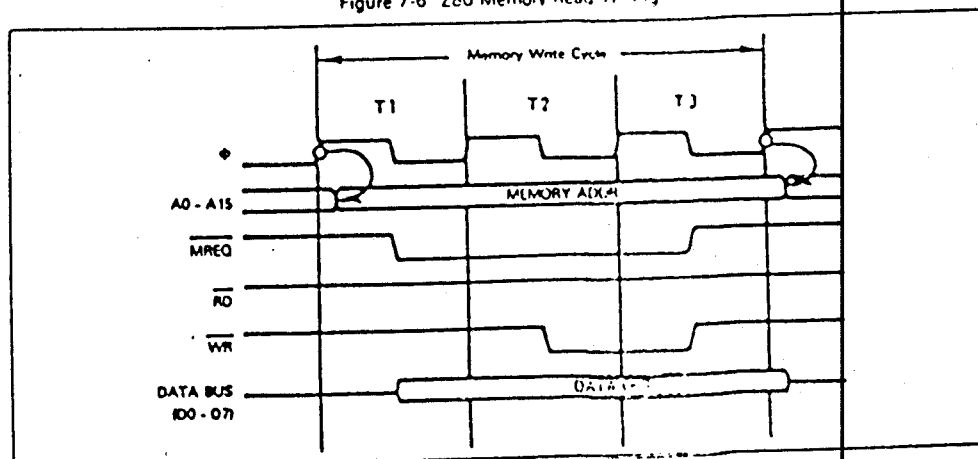


Figure 7-7 Z80 Memory Write Timing

### MEMORY WRITE OPERATION

Figure 7-7 illustrates memory write timing for the Z80. The only differences between memory read and memory write timing are the obvious ones:  $\overline{WR}$  is pulsed low for a write, and high is used as a strobe by memory interface logic to read data off the Data Bus.

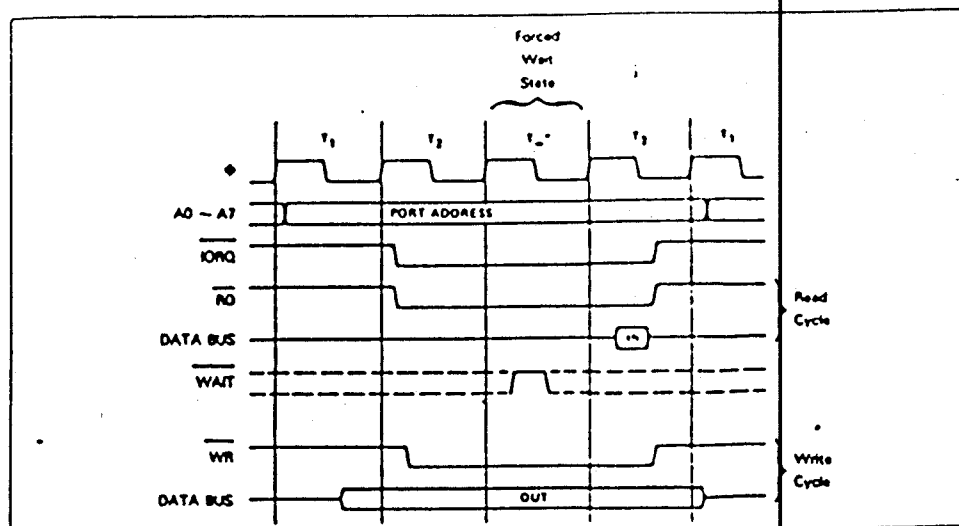


Figure 7-9 Z80 Input or Output Cycles

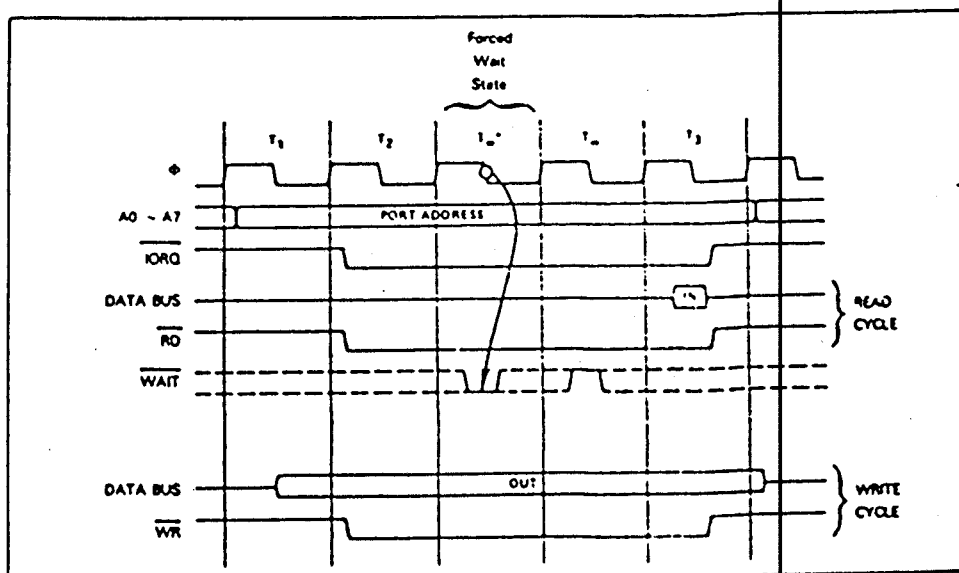


Figure 7-10. Z80 Input or Output Cycles with Wait States

### BUS REQUESTS

The Z80 does not have a Hold state as described for the 8080A, but Z80 bus request logic is equivalent. The Z80 will float Address, Data and tristate Control Bus lines upon sensing a low  $BUSRQ$  signal.  $BUSRQ$  is sampled by the Z80 CPU on the rising edge of the last clock pulse of any machine cycle. If  $BUSRQ$  is sampled low, then tristate lines are floated by the CPU, which also outputs  $BUSAK$  low. The Z80 CPU continues to sample  $BUSRQ$  on the rising edge of every clock pulse. As soon as  $BUSRQ$  is sensed high, floating will cease on the next clock pulse. This timing is illustrated in Figure 7-11.

### THE WAIT STATE

Like the 8080A, the Z80 allows a Wait state to occur between clock periods  $T_2$  and  $T_3$  of a machine cycle. The Wait state frees external logic or memory from having to operate at CPU speed.

The Z80 CPU samples the  $\overline{\text{WAIT}}$  input on the falling edge of  $\Phi$  during  $T_2$ . Providing  $\overline{\text{WAIT}}$  is low on the falling edge of  $\Phi$  during  $T_2$ , Wait clock periods will be inserted. The number of Wait clock periods inserted depends strictly on how long the  $\overline{\text{WAIT}}$  input is held low. As soon as the Z80 detects  $\overline{\text{WAIT}}$  high on the falling edge of  $\Phi$ , it will initiate  $T_3$  or the next rising edge of  $\Phi$ .

Note that the single Z80  $\overline{\text{WAIT}}$  signal replaces the  $\overline{\text{READY}}$  and  $\overline{\text{WAIT}}$  8080A signals. As this would imply, no signal is output telling external logic the Z80 has entered the Wait state. In the event that external logic needs to know whether or not a Wait state has been entered, these are the rules:

- 1) The Z80 will sample  $\overline{\text{WAIT}}$  on the falling edge of  $\Phi$  in  $T_2$ .
- 2) If  $\overline{\text{WAIT}}$  is low, then the Z80 will continue to sample the  $\overline{\text{WAIT}}$  input for all subsequent Wait state clock periods.
- 3) The Z80 will not sample the  $\overline{\text{WAIT}}$  input during any clock period other than  $T_2$  or a Wait state.

Figure 7-8 illustrates Z80 Wait state timing.

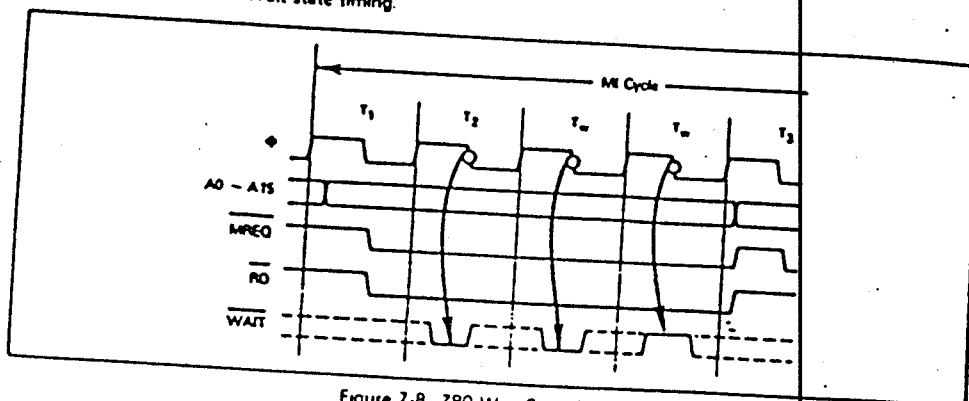


Figure 7-8. Z80 Wait State Timing

### INPUT OR OUTPUT GENERATION

Timing for Z80 input and output generation is given in Figures 7-9 and 7-10.

The important point to note is that Zilog has acknowledged the infrequency with which typical I/O logic can operate at CPU speed. One Wait clock period is therefore automatically inserted between  $T_2$  and  $T_3$  for all input or output machine cycles. Otherwise timing differs from memory read and write operations only in that  $\overline{\text{IORQ}}$  is output low rather than  $\overline{\text{MREQ}}$ .

Note that there is absolutely nothing to prevent you from selecting I/O devices within the memory space. This is something we did consistently in the 8080A chapter when describing 8080A support devices. But if you adopt this design policy, remember that your I/O logic must execute at CPU speed, unless you insert Wait states.

FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO ITS

EE 1799 TUGAS AKHIR - 6 SKS

No.pokok dan Nama Mahasiswa : 2852200363 - C DAMAR SASONGKO  
Bidang Studi : Teknik Komputer  
Tugas diberikan : Awal Semester Genap 1989/1990  
Tugas diselesaikan : Awal Semester Ganjil 1990/1991  
Dosen Pembimbing : 1. DR. Ir. Soepeno Djanali M.Sc  
2.

Judul TUGAS AKHIR : PERENCANAAN DAN PEMBUATAN STAVOLT ELEKTRONIK  
YANG TERPROGRAM .

(Judul harus mempunyai Batas Kualitatif, Sasaran, dan Batas Kuantitatif)

Uraian Tugas Akhir :

Perkembangan teknologi komputer yang sangat cepat dan berkembang secara meluas telah memudahkan manusia untuk mengolah informasi secara cepat dan mudah.

Sebagai salah satu perlengkapan yang sangat mendukung proses pengolahan data, maka stavolt diharapkan dapat untuk menyediakan tenaga yang sestabil mungkin sehingga tidak terjadi kerusakan komputer dan pengolahan data bisa berjalan dengan lancar.

Dalam tugas akhir ini, akan dibahas tentang perencanaan dan pembuatan alat menggunakan minimum system dengan mikroprosesor z-80, untuk pengaturan penjagaan kestabilan tegangan, sehingga jika ada perubahan input, maka kontinuitas proses dan keamanan data dapat terpelihara.

Menyetujui :  
Bidang Studi Teknik Komputer  
Koordinator,

(Dr/Ir. SOEPENO DJANALI, Msc)  
NIP.130368610

Surabaya, 26 Maret 1990

Dosen Pembimbing,

(Dr/Ir. SOEPENO DJANALI, Msc)  
NIP.130368610



Mengetahui  
Teknik Elektro FTI-ITS  
Ketua,

ABDURROBBIL MAHMUDSYAH, M.Eng  
NIP. 130620749

Catatan : - Naskah Tugas Akhir harus diketik dan diajukan rangkap lima  
- Tugas Akhir harus memenuhi ketentuan tata-tulis dan diseminarkan dalam rangka Ujian Tugas Akhir Jurusan Teknik Elektro FTI-ITS.  
- Setelah disetujui, tiga buku Tugas Akhir diserahkan kepada Ruang Baca Jurusan Teknik Elektro FTI-ITS dan PERPUSTAKAAN ITS lewat Panitia Ujian Tugas Akhir Jurusan Teknik Elektro FTI-ITS.

## U S U L A N   T U G A S   A K H I R

A. JUDUL TUGAS AKHIR: PERENCANAAN DAN PEMBUATAN STAVOLT ELEKTRONIK YANG TERPROGRAM.

B. RUANG LINGKUP : - Teknik Pemrograman.  
- Bahasa Assembly.  
- Mikroprosessor

C. LATAR BELAKANG :

- Perkembangan teknologi komputer dalam pengolahan data yang berkembang sangat cepat, telah membantu manusia dalam mendapatkan informasi dengan cepat dan mudah.

- Untuk keamanan dari data-data yang diolah, diperlukan suatu supply tenaga yang stabil, yang bisa bekerja untuk perubahan respons yang cepat.

- Masih kurang akuratnya stavolt yang ada dipasaran, karena mempunyai range perubahan yang terlalu besar + 45 volt, sedangkan step perubahan yang digunakan juga masih terlalu besar + 15 volt.

D. PENELAAHAN STUDI :

Alat yang akan dibuat nantinya, akan mampu untuk menangani perubahan sebesar + 5 volt, sehingga akan mampu menghasilkan tegangan yang stabil.

E. T U J U A N :

Merencanakan dan membuat suatu perangkat keras yang digunakan untuk menstabilkan tegangan yang masuk pada komputer agar tidak terjadi kerusakan komputer akibat drop tegangan.

F. LANGKAH-LANGKAH :

1. Studi literatur untuk mempelajari rangkaian minimum system z-80 , khususnya penggunaan rangkaian ADC .
2. Mempelajari sistem kerja dari transformator yang disulut dengan triac .
3. Perencanaan dan pembuatan alat .
4. Melakukan analisa dan percobaan terhadap alat yang telah dibuat serta melakukan perbaikan bila perlu.
5. Penulisan naskah tugas Akhir

G. JADWAL KEGIATAN :

KEGIATAN	B U L A N							
	I	II	III	IV	V	VI	VII	VIII
1	■	■						
2			■					
3			■	■	■	■	■	
4						■	■	
5							■	■

H. RELEVANSI :

Hasil yang diperoleh dari tugas akhir ini diharapkan dapat membantu pemakai komputer IBM-PC/XT/AT dalam menyediakan supply tenaga yang lebih stabil .